Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Diseño y Desarrollo de Videojuegos

Proyecto Final

Del movimiento 3D a la pedagogía biomecánica

Autor del proyecto: Cristina Berrocal Elu

Director del proyecto: Eduardo Jiménez

Villanueva de Gállego, 16 de Junio de 2021

universidad SANJORGE

GRUPO SANVALERO

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Diseño y desarrollo de Videojuegos por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Si doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma Fecha 16 de Junio de 2020

Dedicatoria y Agradecimiento

Quiero agradecer la confianza depositada por Vanessa Bataller que gracias a su apoyo he podido comenzar este proyecto.

A Eduardo Jiménez tutor y director académico de este proyecto, que con su ayuda profesional y personal he logrado llegar hasta aquí.

Dedico este proyecto especialmente al esfuerzo constante de mi familia que ha permitido que tenga la oportunidad de labrarme el futuro que he elegido. Y a todo ese cariño y apoyo que me dan constantemente para poder evolucionar y crecer como persona y profesional.

También quiero dedicar este proyecto a mi pareja que sin su ánimo y apoyo constante esto no habría funcionado de la misma manera.

También a mis amigas esas que me acompañan desde el inicio hasta el final y que siempre tienen algo esencial que decirte para seguir hacia delante en cualquier momento.

Tabla de contenido

Índice de tablas

Índice de Ilustraciones

Resumen

Este documento recopila el estudio, planificación e implementación del Proyecto Final del Grado de Diseño y Desarrollo de Videojuegos llamado *Del Movimiento 3D a la pedagogía biomecánica*.

El principal objetivo de este Proyecto de Fin de Grado es la realización de una herramienta educativa que sirva de apoyo a la enseñanza de la biomecánica en las aulas, siendo así un software capaz de albergar patrones de movimientos aplicados a la actividad física capturados mediante sistema *Motion Capture* instalado en los laboratorios de la facultad.

El resultado final de este proyecto es una aplicación web multiplataforma publicada en producción, con control de acceso mediante roles, que permite la visualización de movimientos 3D, así como la gestión de estos y su clasificación en la plataforma web.

De este modo el Proyecto de fin de grado (PFG) que nos ocupa, consiste en el diseño e implementación de un conversor, back-end, front-end, y puesta en producción del software para un uso rápido, cómodo y útil de los usuarios, desde cualquier lugar.

Abstract

This document compiles the study, planning, design and implementation of a Final Project for the Videogame Design and Development Degree titled 'From 3D movement to biomechanical pedagogy'.

The main objective of this Final Project is the implementation of an educational tool that supports the teaching of biomechanics in the classroom. With the use of a Motion Capture System provided by the University, our software can host the information retrieved from it and convert it into concrete patterns that can be later views and stored.

The final result of this project is a multiplatform web application published in production, with access control through roles, that allows the visualization of 3D movements, as well as their management and their classification on the web platform.

In this way, this Final Degree Project is composed of the design and implementation of a conversion tool, back-end and front-end, ultimately published in a production environment ready for a fast, usable and comfortable experience from anywhere in the world.

Palabras Clave

Angular, herramienta educativa, NestJs, Repositorio 3D, visor 3D, aplicación web, motion capture, conversor, plataforma docente.

1 Introducción

Este proyecto nace de una idea personal y crece junto al conocimiento de Vanessa Bataller como necesidad al apoyo educativo en el laboratorio de la Facultad de Salud de la Universidad San Jorge. Posteriormente comienza a coger forma con la ayuda de Eduardo Jiménez que orienta el proyecto y lo encamina hacia una estructura real.

En el ámbito educativo existen diferentes plataformas de apoyo para el aprendizaje, como Moodle [1], Edmodo [2] o Classroom [3], estas plataformas sirven como herramientas de gestión de aprendizaje, lo que se busca con este proyecto es ser un apartado o sección viable que tuviese cabida dentro de un tipo de plataformas como las ya nombradas. En este caso el proyecto se diseña como una herramienta educativa individual de apoyo para el profesorado de biomecánica, en una primera perspectiva.

Esta visión evoluciona de tal modo que esta herramienta tiene como labor ser aprovechada por el laboratorio de investigación de Salud de la Universidad San Jorge, siendo útil como repositorio educativo permanente de capturas 3D, accesible remotamente.

Esta actual perspectiva ayuda a tener una mayor muestra de movimientos 3D en el laboratorio para futuras investigaciones y al apoyo educativo en el aula.

Para poder crear este proyecto completamente se necesita una base de datos, una interfaz intuitiva y acceso remoto, así como un conversor de formatos. Por lo que hay que diseñar, desarrollar e implementar un back-end, fron-end y un conversor, y conectarlos y desplegarlos en producción.

A la hora de fundamentar esta idea se realiza una encuesta con la herramienta Google Forms [4] para tener un mayor conocimiento de las necesidades reales de futuros usuarios de la plataforma.

La encuesta se puede encontrar en el Anexo IV – Encuesta. Esta encuesta ha sido respondida por un total de 35 usuarios del sector sociosanitario, de estos 35 usuarios los estudios cursados son los siguientes:

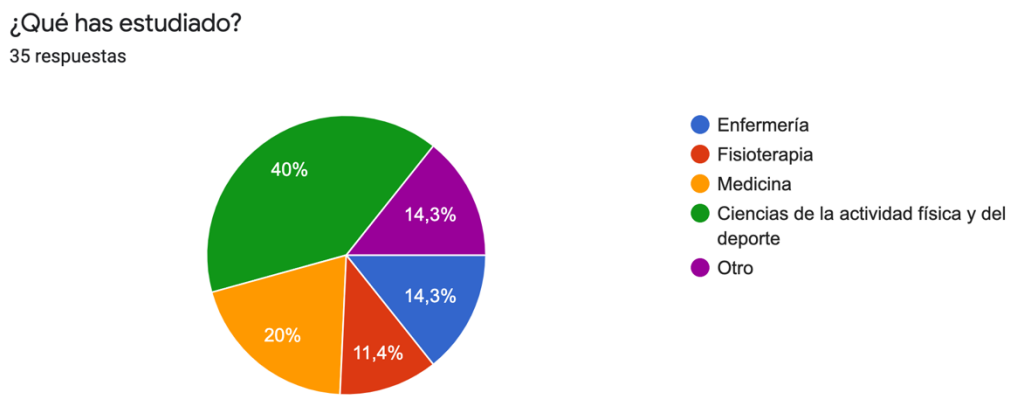


Ilustración 1 Gráfica de estudios por usuario de la encuesta

De la totalidad de estos usuarios el 40% se dedica a la educación, de los cuales el 14.2% se dedica a la enseñanza universitaria.

El 71.4% de los usuarios admiten tener algún conocimiento sobre la biomecánica, y el 54.3% haberla estudiado como asignatura en alguna ocasión de su vida, de estos es importante destacar que el 85,7% de los usuarios cree que un repositorio o biblioteca de patrón de movimientos básicos podría haber ayudado a su proceso de enseñanza-aprendizaje sobre la biomecánica. Y el 100% cree que tener acceso a un repositorio 3D de movimientos puede ser útil para el proceso de enseñanza-aprendizaje en el ámbito deportivo y de la salud.

El 80% dice que utilizaría esta idea para el proceso de enseñanza-aprendizaje en el ámbito del deporte y la salud y el 20% que tal vez lo haría.

Como aportaciones significativas se han respondido las siguientes:

- "Sería muy interesante una biblioteca de imágenes de los movimientos técnicos de deportistas de élite para poderlos proyectar y luego poder hacer comparativas de los gestos técnicos de unos y otros."
- "Opino que la biomecánica aporta gran valor en la técnica física y deportiva, ayudando a los deportistas a ser más eficientes en la ejecución de sus movimientos. Creo que esa herramienta ayudaría bastante a entender la materia."
- "Hay muchas personas, tanto profesionales como pacientes de fisioterapia, que tienen una visión espacial muy limitada, por lo que son incapaces de imaginar y realizar un movimiento explicado de viva voz. Es por ello por lo que un repositorio de movimientos

3D resulta muy interesante tanto para la comprensión personal durante los años de estudio como para la aplicación en terceros en el ámbito laboral."

Gracias a esta encuesta se ha podido valorar la utilidad de este proyecto ya que se pone en manifiesto que puede ser útil y sirva para mejorar la calidad de la enseñanza en la biomecánica. También cabe decir que se ha tenido en cuenta los comentarios y sugerencias de manera significativa ya que se han implementado o situado dentro del plan de mejora.

2 Antecedentes y estudio del Arte

En cuanto a los antecedentes a este proyecto hay varias partes a estudiar. En primer lugar, el contexto del proyecto para explicar de dónde partimos, qué tenemos y qué queremos conseguir.

2.1 Contexto

Desde un inicio se quiere plantear este proyecto como un proyecto en Angular [5] con la librería Three.js [6].

Se selecciona estas tecnologías debido a un conocimiento previo sobre Angular y Three.js aunque nunca en un mismo proyecto. El deseo de aunar todas estas tecnologías en un solo proyecto y obtener más experiencia en su uso fue un elemento clave en su elección. El reto en este caso es poder llevar a cabo esta idea con tecnología Angular como front-end y NestJs como back-end.

También se tuvo en cuenta la biblioteca Model-viewer [7] , pero se descartó, ya que la carga de objetos no era tan variada como la de Three.js.

¿Qué tenemos?

En este punto se comienza a investigar sobre los diferentes softwares utilizados en las instalaciones de la Universidad San Jorge (USJ) para la captura de movimientos, es importante estudiarlos ya que los formatos de archivo que pueden exportar las herramientas disponibles condicionarán el uso de la librería Three.js además de poder exigir el desarrollo de herramientas propias para poder realizar conversiones entre los formatos producidos por las herramientas y los aceptados por Three.js.

En el laboratorio de la facultad de Salud de la USJ hay a disposición, bajo supervisión, el sistema digital de fotogrametría SMART-D de BTS Bioengineering [8], el sistema de acelerometría Noraxon Myomotion Clinical IMU [9] y el sistema de acelerometría Wearnotch [10]. A continuación, se detallará brevemente el funcionamiento de estos tres sistemas.

2.1.1 Sistema de análisis de movimiento BTS SMART-D

El sistema BTS SMART-D de la marca BTS Bioengineering, es un sistema digital optoléctrico de alta precisión con marcadores pasivos para el análisis y estudio de cinemática del

movimiento. Es un sistema que sirve para capturas de movimiento basado en tecnología óptica por infrarrojos, diseñado para uso biomecánico y clínico, este sistema puede ejecutar la reconstrucción tridimensional de trayectorias de un cierto número de pequeños marcadores reflectantes en el cuerpo del paciente, mediante un conjunto de cámaras colocadas alrededor del volumen donde se realizará la acción. La posición de las cámaras se definirá gracias al paso de calibración. [11]

En la siguiente imagen podemos ver una representación de captura de movimiento con una versión nueva del sistema SMART-D, la versión SMART-DX, esta versión no está disponible en las instalaciones, pero muestra un diagrama claro del funcionamiento de este sistema.

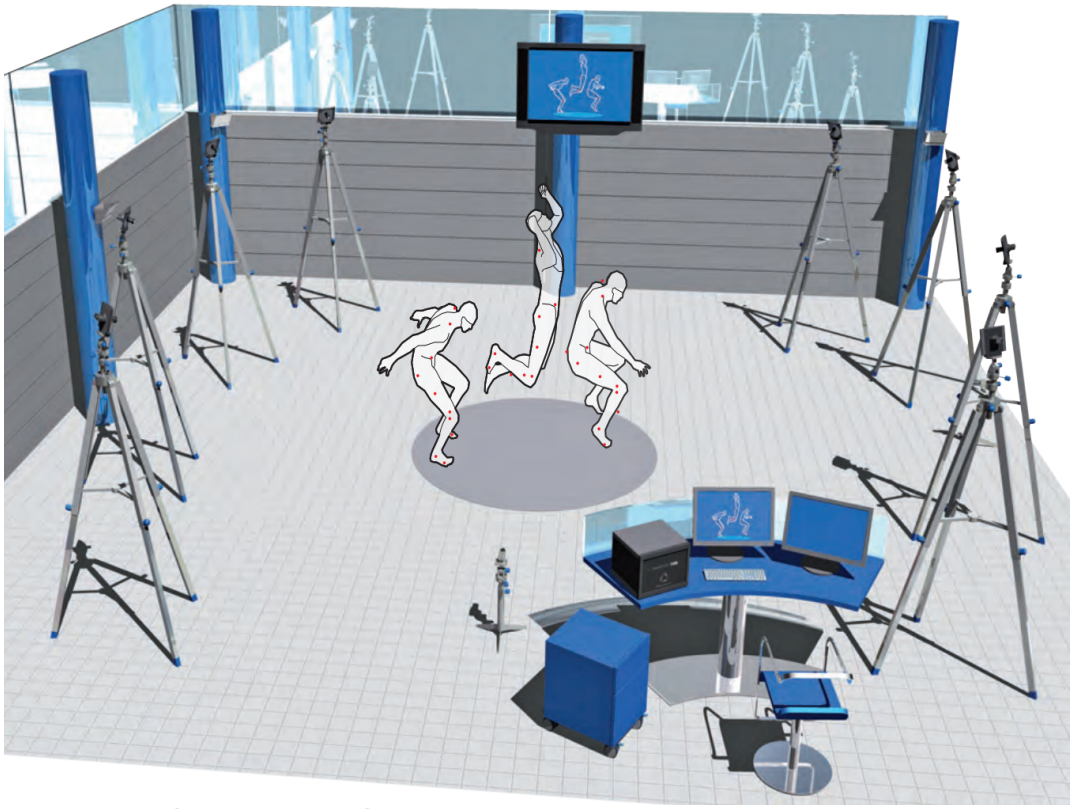


Ilustración 2 Representación de captura del movimiento del sistema SMART – DX, [12]

Este sistema tiene diferentes formatos de exportación para las capturas de movimiento, son las siguientes:

- Archivos nativos:
    - **SMART System Data Files (*.TDF).** Contienen pistas 3D, datos de la plataforma de fuerza, señales analógicas electromiográficas e información de enlaces de modelos.

- **SMART analyzer Protocol Files(*.EPX).** Los protocolos y los informes correspondientes se almacenan en formato de archivo EPX.
- **Archive Files (*.MDX).** Estos archivos se utilizan para almacenar todos los datos contenidos en una adquisición: es decir, datos leídos del archivo original y datos creados por operadores de protocolo. Este tipo de archivo sigue un formato XML.
- **Normative Bands Files(*.ENB).** Un archivo de Bandas normativas contiene información sobre el rango normal de un dato. En el análisis clínico, los archivos de bandas normativas contienen información sobre el rango normal de valor biomecánico (como la rotación de los ángulos articulares, las dimensiones de los segmentos corporales, etc.) y se utilizan a menudo para comparar datos de un sujeto en particular con los objetos normales.

- Archivos de importación/exportación:
  - **Archivos C3D (*.C3D).** El C3D es un formato de archivo de dominio público que se usa ampliamente en biomecánica, animación y análisis de la marcha. Puede encontrar más información al respecto en el sitio web oficial de C3D [13].
  - **Text Files (*.EMT).** SMARTanalyzer exporta datos en un archivo de prueba con la extensión EMT. Los archivos de texto exportados por SMARTanalyzer son legibles por Excel.

Con el sistema BTS en las instalaciones de la universidad se consiguen capturas del tren inferior y tren superior en (*.mdx) de las siguientes características:

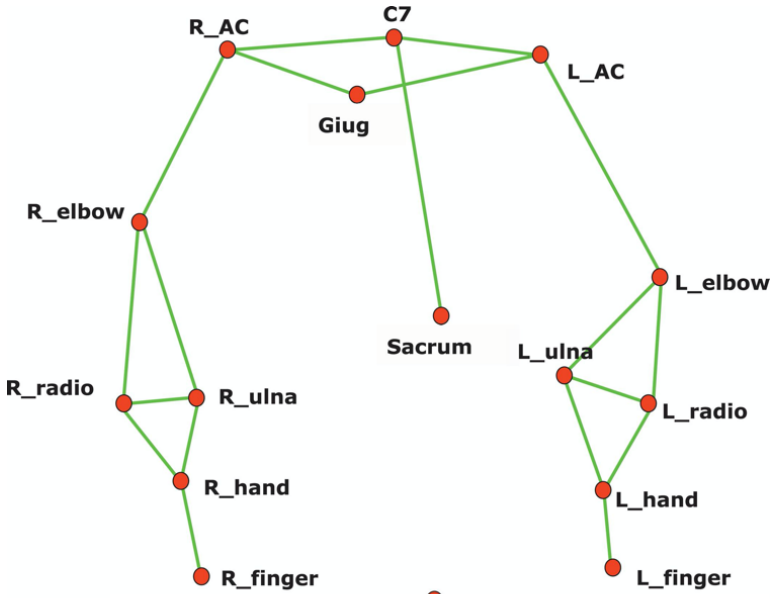


Ilustración 3 BTS - Modelo anatómico tren superior

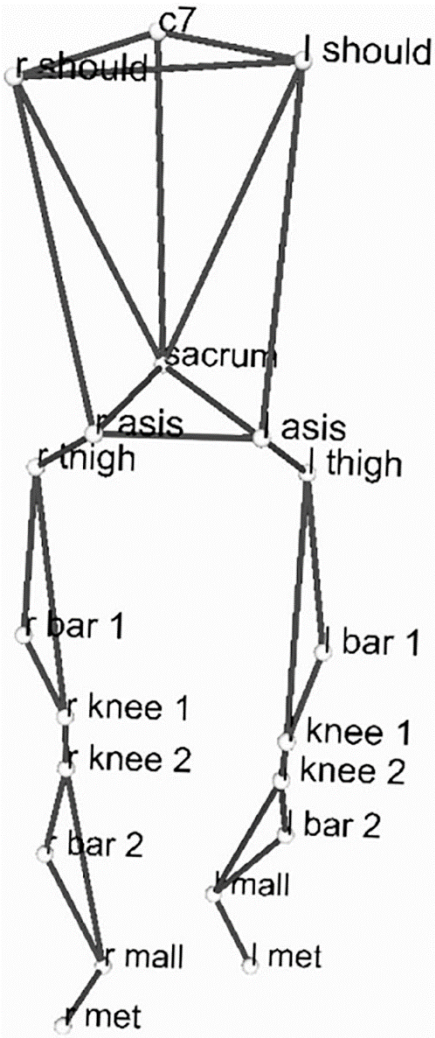


Ilustración 4 Modelo anatómico BTS tren inferior

Este tipo de sistema tiene como ventaja una alta precisión, debido al número de cámaras que se utilizan y a ser un sistema óptico calibrado previamente a cada uso para conseguir una elevada precisión.

2.1.2 Noraxon Myomotion Clinical IMU

El sistema NORAXON Myomotion y el software es Myomotion Clinical IMU hace uso de la acelerometría. El sistema de análisis y captura de movimiento 3D de Noraxon consta de un conjunto de 1 a 16 sensores que utilizan la tecnología de unidad de medición inercial (IMU). Mediante "algoritmos de fusión", la información del sistema que consta de un acelerómetro 3D, un giroscopio y un magnetómetro se utiliza para medir los ángulos de rotación 3D de cada sensor

en el espacio absoluto ("yaw-pitch-roll", también conocido como ángulos de orientación o navegación). [14]

Ilustración 5 Captura de pantalla del sistema Noraxon [15]

Puede exportar registros MR3 en formatos nativos dentro de su versión en ejecución, tales como:

- **Archivo CSV**. Los datos numéricos se pueden exportar a formato CSV compatible con Excel.
- **Archivo C3D**. Los datos numéricos (señales) se pueden exportar a formato C3D [13].
- **Archivo Excel (*.slk).** Los datos numéricos (señales) se pueden exportar a formato SLK compatible con Excel.
- **Archivo MatLab (*.mat).** Los datos numéricos (señales) se pueden exportar al formato MatLab.
- **Exportar datos de myoMOTION a Biovison BHV**. Exportación de datos de MyoMotion (se requiere cuerpo completo) a BHV.

2.1.3 Wearnotch

Notch es una plataforma para aplicaciones y productos de análisis de movimiento. El componente de hardware es la red de sensores inerciales portátiles, que reconstruye los avatares en 3D de los usuarios a partir de sus movimientos directamente en sus teléfonos inteligentes. El componente de hardware, también llamados Notch ver Ilustración 6, son el motor de inteligencia artificial para el procesamiento del movimiento, este hardware complementado con las aplicaciones de software son en su conjunto la esencia de esta tecnología.

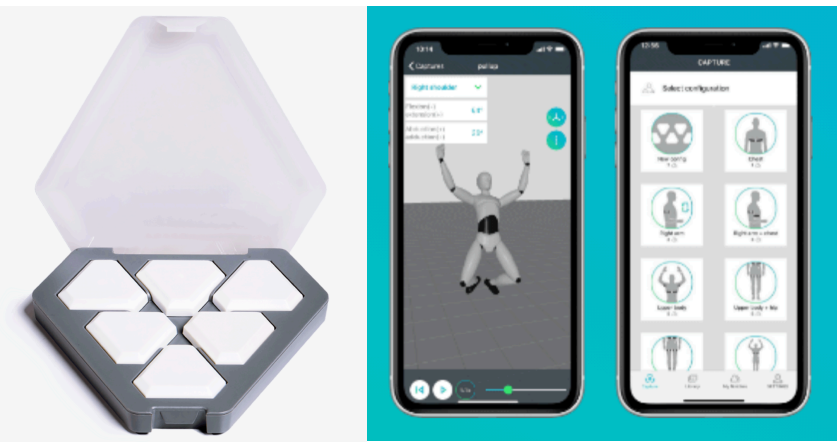


Ilustración 6 Notch

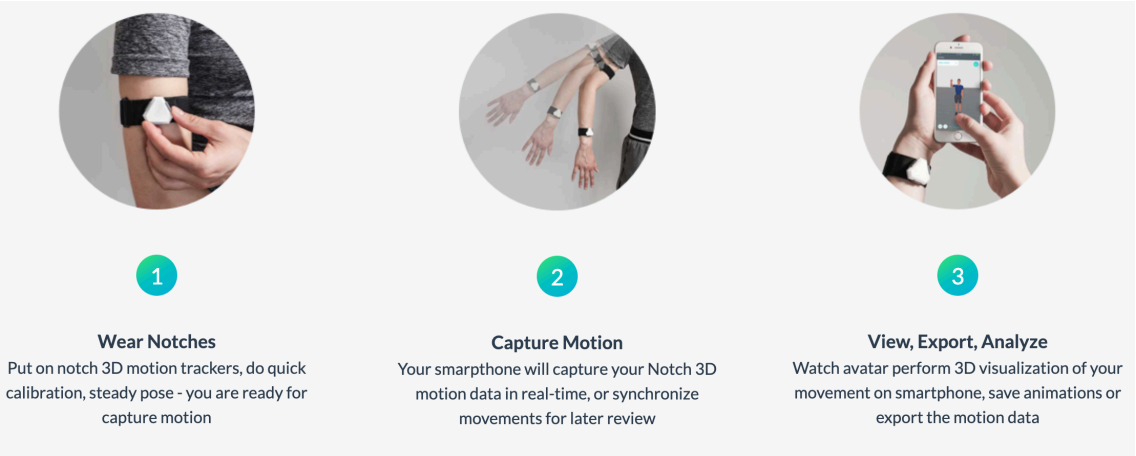


Ilustración 7 Wearnotch especificaciones

Los Notch son pequeños dispositivos que pueden formar una red de sensores inalámbricos de 2 o más nodos, como se puede ver en la Ilustración 7. Cada módulo está equipado con 3 sensores MEMS de alta precisión, que incluyen un acelerómetro, un giroscopio y una brújula. La memoria integrada permite la captura de movimiento sin una conexión constante a un teléfono inteligente. Las muescas utilizan Bluetooth Low Energy para comunicarse con el teléfono inteligente.

Las exportaciones posibles de estas capturas son:

- **CSV,** (comma-separated values) este formato es un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.
- **BVH**, [16] Los archivos de Biovision BVH Files contienen información de la jerarquía del esqueleto además de los datos de movimiento. Es un formato común y estandarizado que se usa en la captura de movimiento, así como en la animación de personajes.

- **Fbx**, es un formato de archivo patentado desarrollado por Kaydara y propiedad de Autodesk desde 2006. Se utiliza para proporcionar interoperabilidad entre aplicaciones de creación de contenido digital. El formato de archivo FBX admite propiedades relacionadas con la geometría y la apariencia, como el color y las texturas. También admite animaciones esqueléticas y morphs. Se admiten archivos binarios y ASCII.

2.1.4 Conclusión de sistemas

Estos softwares y sistemas son de pago y nos ofrecen un visor de esqueleto con toda la información relativa a los datos de movimiento ya sea dirección, velocidad, fuerza, etc. También nos ofrecen la oportunidad de ver gráficas relacionadas a estos datos. El problema que tienen estos softwares es que son muy complejos y necesitan de una licencia de pago para ser utilizados por todo el personal docente. Por ello solo son empleados en dispositivos propios de la universidad y bajo docentes con permiso y licencia. Esto restringe mucho su uso para la docencia.

Si bien es cierto que estos softwares brindan una gran cantidad de información que podría ser relevante para la formación académica, entre otras muchas cosas, este no es su uso principal. Por este motivo desde el laboratorio de la facultad de salud de la USJ surge la idea de crear un repositorio como base de datos de todos aquellos movimientos capturados mediante estos sistemas, accesible para toda la comunidad universitaria de la Universidad San Jorge.

Estos softwares pueden exportar el movimiento en diferentes formatos. Esto ayuda a poder crear un visor online de una biblioteca de movimientos. El problema de estos formatos es que no todos son legibles en todas las tecnologías. Los formatos que nos interesan y podemos tratar con la biblioteca Three.js son los siguientes:

- BVH
- FBX
- Gltf

No todos estos formatos, con los que, si podemos tratar con el stack tecnológico seleccionado, son formatos exportables desde los sistemas instalados en la universidad, algunos de ellos coinciden como es el caso de FBX o BVH en Wearnotch. Esto ocasiona la necesidad de tener que generar un conversor de formatos. Se toma la decisión de que el conversor sea al formato FBX, que es el formato más cómodo de usar para la librería Three.js. Por lo tanto, se convertirá el formato mdx de BTS a FBX.

2.2 Marco tecnológico

En este momento la necesidad de un repositorio online implica una web disponible desde cualquier lugar, con un front-end, y back-end disponibles en un hosting en producción, y un conversor de formatos a FBX.

Como premisa de este PFG para el desarrollo del front-end se selecciona la tecnología Angular, la tecnología NestJs [17] para el back-end y la librería Three.js para el visor de movimientos. Por otro lado, y gracias a los consejos y la ayuda del profesor Eduardo Jiménez se decide utilizar Unity [18] para la realización del conversor. A nivel de servidores y puesta en producción se selecciona un Servidor Virtual Privado (VPS) de la empresa OVH [19] y un hosting de OVH [20], con implantación de Apache2 [21] y Pm2 [22].

Esta selección de tecnologías tiene en parte motivo por la gran comunidad de apoyo en foros y chats de las propias webs que facilitan ayuda con la mayoría de los problemas que se pueden encontrar durante el desarrollo.

A continuación, se detallan y describen brevemente las herramientas utilizadas y la interacción interna necesaria entre ellas para el desarrollo de este PFG. Las herramientas utilizadas son las siguientes:

Desarrollo	Tecnología
Herramienta diseño	Draw.io
Herramienta de desarrollo	Visual Studio Code
Front-end	Angular 8
Back-end	NestJs + Node.js
Documentación back-end	Swagger
Base de datos	MySQL
Visor movimientos	Three.js
Conversor	Unity
Servidores	VPS OVHCloud
Puesta en producción del back-end	PM2
Puesta en producción del front-end	Apache2
Gestión proyecto	Trello
Gestión de versiones	Git – GitHub - Sourcetree

Tabla 1 Tecnologías seleccionadas

2.2.1 Draw.io

Draw.io o Diagrams.net es un software de diagramas en línea gratuito para hacer diagramas de flujo, diagramas de procesos, organigramas, UML, ER y diagramas de red.

2.2.2 Visual Studio Code

Visual Studio Code es un editor de texto plano desarrollado por Microsoft totalmente gratuito y de código abierto para ofrecer a los usuarios una herramienta de programación avanzada, incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código

2.2.3 Angular 8

Angular [23] es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo-Vista-Controlador. En este Proyecto se ha usado Angular en la versión 8.

2.2.4 NestJS

NestJS es un proyecto de código abierto con licencia del MIT. Es un framework sobre Node.js con lenguaje TypeScript que sirve para crear aplicaciones del lado del servidor, abstrae la utilización de Express y Socket.io a través del uso de decoradores, para definir y extender el funcionamiento de los diferentes componentes del sistema, utilizándolos también como base para definir la estructura de la aplicación y permitiendo adaptar funcionalidad reutilizable y proporcionada por el framework de manera muy clara y legible. Permite modularizar las aplicaciones aplicando conceptos de orientación a objetos y programación funcional y reactiva.
La documentación oficial se puede encontrar aquí [24] y el repositorio oficial [25].

2.2.5 Node.js

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos [26]. Se utiliza principalmente para servidores controlados por eventos y sin bloqueo. Permite a los desarrolladores de software iniciar tanto el front-end como el back-end de las aplicaciones web.

2.2.6 Swagger

Con Swagger UI puedo organizar los métodos, utiliza un documento JSON o YAML existente y lo hace interactivo. Crea una plataforma que ordena cada uno de nuestros métodos (get, put, post, delete) y categoriza las operaciones.
Swagger es un conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful. Fue desarrollado por SmartBear Software [27] e incluye documentación automatizada, generación de código, y generación de casos de prueba.

El servicio web RESTful, está basado en REST (REST - Representational State Transfer) define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia los clientes en diversos lenguajes. Usualmente los RESTful web service tienen estas características:

- utiliza los métodos HTTP de manera explícita
- no mantiene estado
- expone URIs con forma de directorios
- transfiere XML, JavaScript Object Notation (JSON), o ambos

Los métodos HTTP que se utilizan son los siguientes:

- Listar y leer: Usan el método GET
- Crear: Usan el método POST
- Actualizar: Usan el método PATCH para actualizar y PUT para reemplazar.
- Borrar: Usan el método DELETE

Para más información sobre RESTful [28].

2.2.7 MySQL

MySQL es un sistema de gestión de bases de datos relacional y sirve para almacenar y administrar datos en bases de datos relacionales utilizando diferentes tablas, vistas, procedimientos almacenados, funciones, etc.

2.2.8 Three.js

Es una biblioteca con lenguaje JavaScript para crear y mostrar gráficos animados en 3D en un navegador Web.

2.2.9 Unity

Unity es un motor de videojuego multiplataforma creado por Unity Technologies.

2.2.10 VPS de OVHcloud

El Servidor virtual privado (VPS), es una partición virtual dentro de un servidor físico que le asigna recursos exclusivos a cada partición. En este caso he seleccionado un VPS de OVH [29]. OVHcloud es un proveedor de alojamiento web y telecomunicaciones francés.

2.2.11 PM2

PM2 es un administrador de procesos para el tiempo de ejecución para Node.js, de código abierto. PM2 también permite mantener las aplicaciones activas para siempre, las vuelve a cargar sin tiempo de inactividad y le ayuda a administrar el registro, la supervisión y la agrupación en clústeres de las aplicaciones.

2.2.12 Apache2

Apache es un servidor web HTTP de código abierto y multiplataforma que implementa el protocolo HTTP/1.12, y la noción de sitio virtual.

2.2.13 Trello

Trello [30] es un software de administración de proyectos con interfaz web, basado en la metodología ágil Kanban, se usa como herramienta de gestión de proyectos. Consiste en una serie de tarjetas, listas y tableros para facilitar la organización del proyecto y poder etiquetar y priorizar las tareas con mayor facilidad. Esta herramienta ha sido utilizada para la gestión y organización de las tareas diarias de este PFG.

2.2.14 Git

Git [31] es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

2.2.15 GitHub

Github [32] es una plataforma de desarrollo colaborativo de software que sirve para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

2.2.16 Ng Zorro

Ng Zorro es una librería de diseño para Angular soportado por Alibabá, posee una extensa cantidad de componentes de UI funcionales basados en Angular y Ant Design.

2.3 Interacción interna

Para ilustrar el marco tecnológico y su funcionamiento e interacción interna se han realizado las siguientes imágenes ilustrativas:

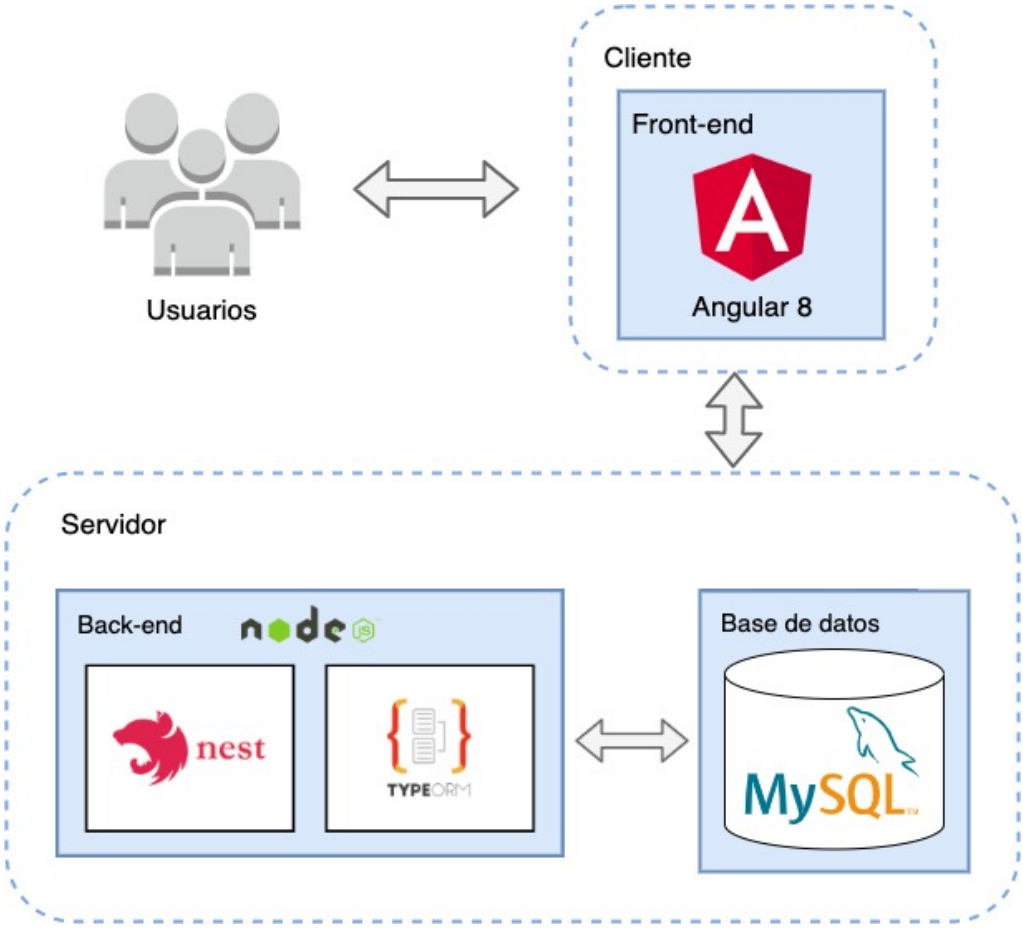


Ilustración 8 Interacción interna del stack tecnológico

El proyecto back-end de la aplicación. Está implementado un servidor Node.js / NestJS que está conectado a una instancia de base de datos MySQL.

En este proyecto se trabaja en una API Node.js simple que funciona con una base de datos MySQL para el almacenamiento de datos y se ha configurado algunas herramientas a su alrededor para facilitar su desarrollado.

Esta es la configuración:
NestJS y typeORM están conectados a una base de datos MySQL. Se construye una API sobre Node.js en NestJs. Es una plataforma bastante flexible y está construida sobre los principios de ExpressJs [33].

3 Objetivos

En este apartado se presentan los objetivos de este PFG. Los objetivos iniciales se especificaron en la propuesta del proyecto adjunta en el Anexo I.

Este Proyecto de fin de grado tiene como objetivo el desarrollo de una aplicación informática que automatice el proceso de conversión y adaptación de ficheros de captura de movimiento 3D a un formato concreto, dicho formato será archivado en una base de datos. También se desarrollará la aplicación web que permita la visualización de estos ficheros junto con las herramientas necesarias para su uso tanto de forma pedagógica como para la investigación.

Los objetivos del proyecto son:

- Creación del sistema de conversión de ficheros de mo-cap (motion capture) a otros que puedan ser visualizados fácilmente en web.
- Creación de una base de datos para su almacenamiento y automatización del proceso para añadir estos ficheros conforme se van creando.
- Creación de una web con una parte de administrador para poder gestionar fácilmente la base de datos.
- Creación de una herramienta web que permita la visualización e interacción con las grabaciones de movimiento 3D para uso pedagógico o de investigación.

4 Metodología

Este proyecto es elaborado y compaginado a la vez que un trabajo a jornada completa y otras asignaturas de diferente índole, por lo que el tiempo dedicado no puede ser una jornada completa, ni puede ser un horario regulado. El tiempo dedicado a este proyecto será parte del tiempo libre y tiempo destinado directamente a esta labor.

Teniendo en cuenta las características aquí descritas se decide optar por una auto organización del tiempo siguiendo algunas de las características de las metodologías ágiles, que detallaremos a continuación. Por lo tanto, podemos decir, que la metodología usada en este PFG es una adaptación propia de las metodologías ágiles a las necesidades del proyecto.

4.1 ¿Cómo funciona la metodología propuesta?

La metodología que se ha seguido en este trabajo ha sido la siguiente:

- Primero se han definido las etapas de trabajo y los objetivos a conseguir en cada etapa, esta fase puede parecerse al backlog de la metodología ágil de SCRUM.
- Posteriormente se han organizado estas etapas de tal manera que la que va segunda ya tenga preparado lo necesario de la primera y así sucesivamente, esto se puede asemejar a los sprint de la metodología ágil SCRUM.
- Las herramientas seleccionadas para gestionar el proyecto y seguir una metodología ha sido la plataforma *on-line* Trello [30] y Excel [34].

4.1.1 Trello

El grueso de las tareas es especificado y ordenado en la herramienta *Trello*, para así poder ver de un vistazo el trabajo a realizar. Las etiquetas de Trello han sido personalizadas para adecuarlas a este proyecto. Quedando de la siguiente manera:

Ilustración 9 Etiquetas personalizadas de Trello

Esta serie de etiquetas clasifican la tarea en diferentes temas. Ya que hay diferentes apartados:

- PRO, referente a la puesta en producción.
- Memoria, referente a la memoria que nos ocupa.
- Front, referente a las tareas del front-end.
- Investigación, referente a las tareas de investigación.
- Conversor, referente al conversor.
- Back, referente a las tareas de back-end.

En el *Anexo V – Tareas de Trello* se pueden visualizar imágenes del panel de Trello.

4.1.2 Tareas en Excel

Una vez entrado en el desarrollo del trabajo las tareas son detalladas en una hoja de Excel, a pesar de que Trello cumple con estas funcionalidades, por decisión del desarrollador y costumbre se prefiere utilizar Excel para este fin.

En esta hoja se detallará la duración, el tipo, el estado y su prioridad, así como la descripción y el objetivo que se busca con cada tarea. Esta hoja Excel será actualizada en todo momento y se incluirán todas aquellas tareas necesarias para cumplir con su enfoque principal "no olvidarse nada en ningún momento", ya que el desarrollo de este proyecto se va realizando en tiempo libre suelen pasar varios días entre horas de desarrollo por lo que es importante dejar siempre detallado el estado de desarrollo del proyecto.

Las características de esta hoja tienen un código de colores y nomenclatura específica que viene detallada a continuación.

TIME	TYPE	STATUS	PRIORITY
LARGE	BUG	PROGRESS	URGENT
MEDIUM	IMPROVEMENT	DONE	MEDIUM
SHORT			NO WORRIES
IMMEDIATLY			IMPROV

Ilustración 10 Características de cada tarea en Excel

4.1.2.1 Duración de las tareas

Las tareas pueden ser consideradas:

- Large, estas tareas puedes ser consideradas larga duración debido a que no se sabe la duración de la tarea y puede ser muy grande y larga. Tareas costosas que pueden implicar varias horas de trabajo.
- Medium, estas tareas pueden ser consideradas largas, pero no demasiado, optimas para una tarde entera.
- Short, las tareas cortas son aquellas que pueden tardar unas pocas horas y no deben tener mayor complicación.
- Immediatly, son aquellas tareas que son prácticamente inmediatas, suelen estar destinadas a retoques en concretos de maquetación o asuntos que ocupan solo unos minutos, pero son suficientemente importantes como para ponerlos y tenerlos en cuenta.

4.1.2.2 Tipo de tareas

Hay dos tipos de tareas las *BUG* y las *IMPROVEMENT*, como su propio nombre indica las que son *IMPROVEMENT* son tareas destinadas a la mejora o incorporación de nuevas características (features) del proyecto, y los *BUG* a la solución de errores surgidos o encontrados.

4.1.2.3 Estatus de cada tarea

El estado de cada tarea puede ser PROGRESS y subrayada en color amarillo, que significará que esta tarea está en proceso de desarrollo y ya ha sido comenzada, o DONE que en su caso ya estará finalizada y en color verde. Las tareas no comenzadas simplemente estarán en blanco.

4.1.2.4 Prioridad de cada tarea

La prioridad de cada tarea es muy importante ya que es la que manda sobre el orden de tareas a ejecutar. Siempre se comenzará por las tareas URGENT (urgentes) coloreadas en naranja.

Posteriormente se seguirá con las tareas MEDIUM (medio) que suelen ser tareas destinadas a la mejora o al funcionamiento base del desarrollo. Las tareas NO WORRIES (no importa), son aquellas tareas que si no se acaban no afectan al funcionamiento del programa pueden ser tareas más relacionadas con el diseño y la apariencia. Finalmente, las tareas IMPROV, son aquellas tareas destinadas a la mejora del software pero que en sí mismas no son necesarias para el funcionamiento del mismo.

4.2 Seguimiento del desarrollo

4.2.1 Diseño de la idea

Primero se realizó un diseño y boceto de la idea mediante la plataforma Draw.io, que fue aprobado por el cliente, en este caso la profesora Vanessa Bataller. Este pasaría a ser el diseño principal del frontal de la web.

Posteriormente a esta idea se elaboró una planificación de tareas que se seguirían para su cumplimiento y seguimiento.

4.2.2 Diagrama de trabajo

Para hacer el seguimiento de las tareas y ver su implicación en horas de trabajo se ha realizado un diagrama de Gantt personalizado para reflejar esta información. En este diagrama se documenta el tiempo dedicado a cada tarea de una forma general, ya que no se han dispuesto de jornadas de 7 horas en la mayoría de los días. Esta tabla representa la jornada de 35h semanales del desarrollador.

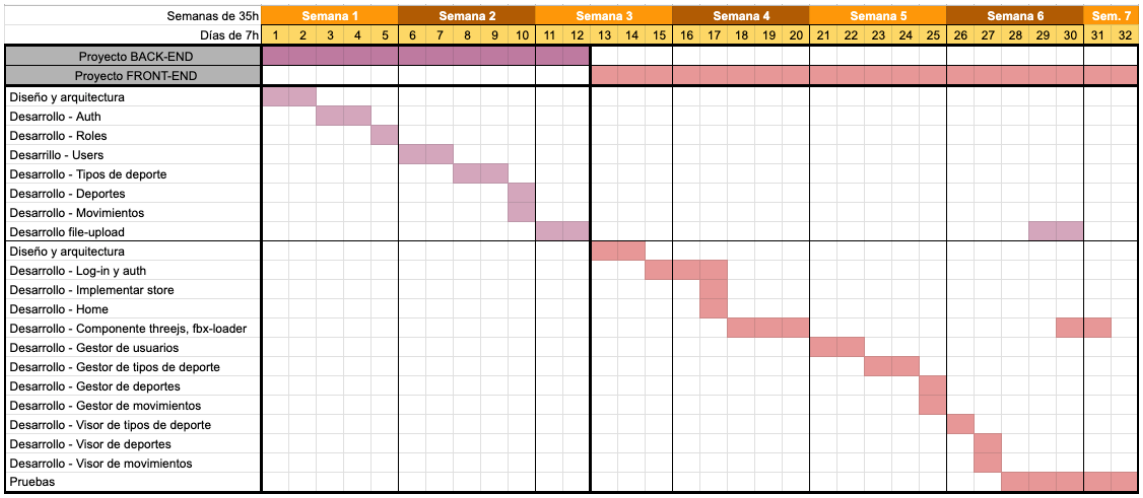


Ilustración 11 Computo de horas

5 Análisis y diseño

En el presente apartado se detalla el análisis de este proyecto, el diseño del front-end, del back-end, y del conversor.

5.1 Análisis del problema

La elaboración de este proyecto implica la problemática de acceso a los datos de la plataforma, en este aspecto nos encontramos con dos actores:

- Administrador
- Usuario

Estos dos actores tendrán diferentes funcionalidades que vemos en detalle a continuación.

5.1.1 Administrador

El administrador será el encargado de gestionar la web, sus funciones serán las siguientes:

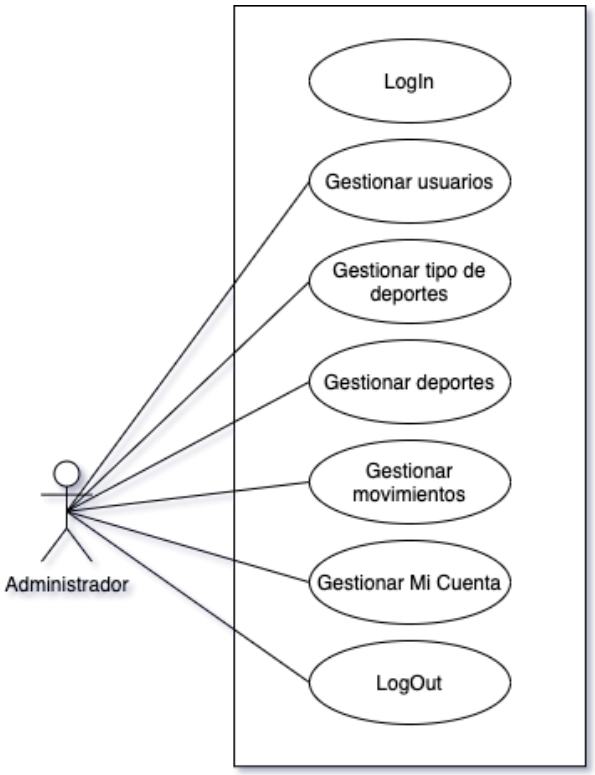


Ilustración 12 Diagrama de caso de uso – Administrador

- Gestionar usuarios: para poder controlar el acceso y roles de la web el administrador podrá gestionar los usuarios realizando los siguientes casos de uso:

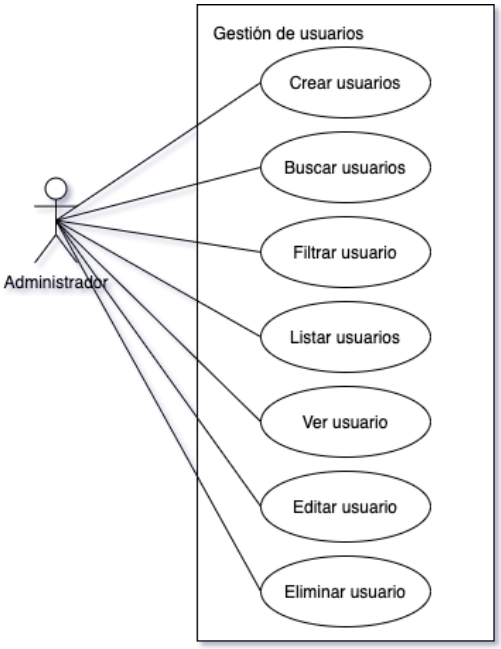


Ilustración 13 Diagrama de caso de uso - Gestor de usuarios como administrador

- Gestionar tipos de deporte: para poder gestionar los tipos de deportes el administrador puede realizar los siguientes casos de uso:

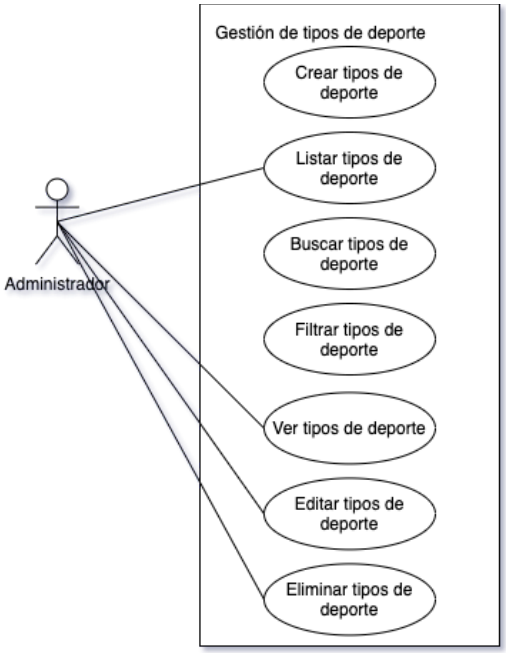


Ilustración 14 Diagrama de caso de uso - Gestión de tipo de deportes

- Gestionar deportes: para gestionar los deportes el administrador puede realizar los siguientes casos de uso:

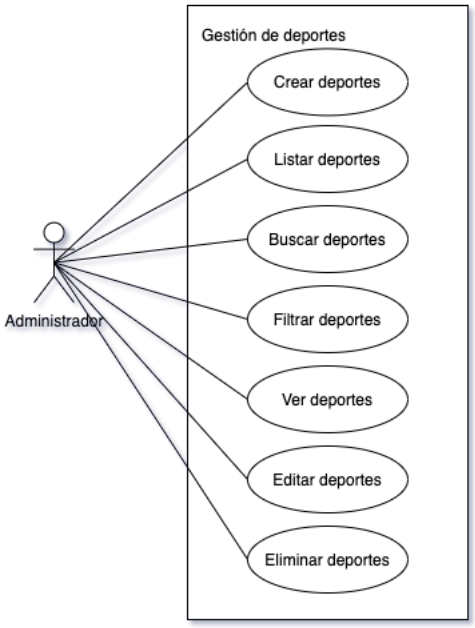


Ilustración 15 Diagrama de caso de uso - Gestión de deportes

- Gestionar movimientos: para gestionar los movimientos el administrador puede realizar los siguientes casos de uso

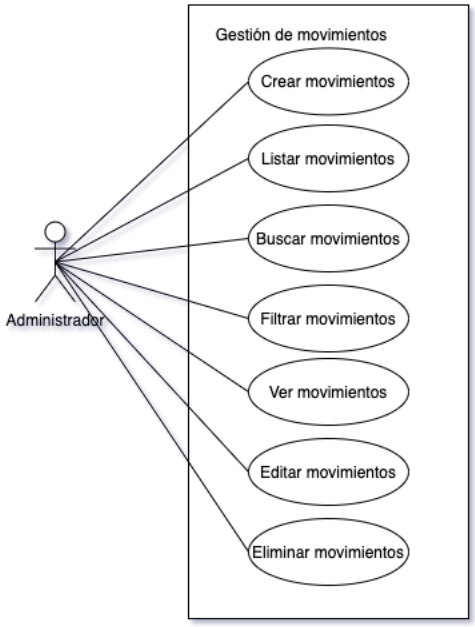


Ilustración 16 Diagrama de caso de uso - Gestión de movimientos

5.1.2 Usuarios

Los usuarios tienen menos acceso a la plataforma ya que no pueden gestionar nada más que su propio perfil, son espectadores, pueden realizar los siguientes casos de uso:

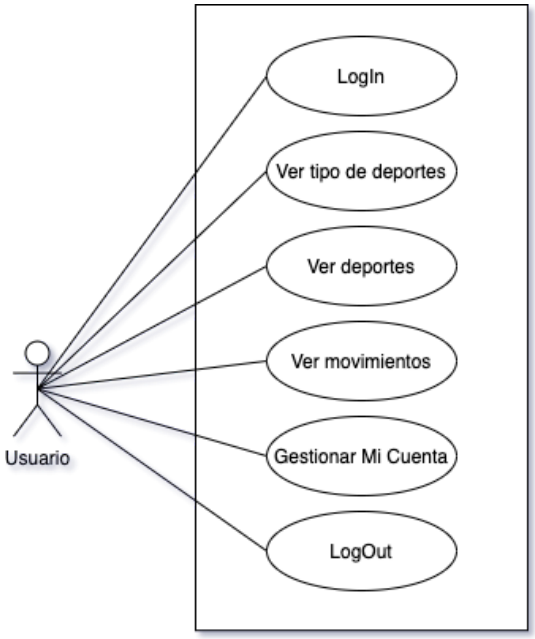


Ilustración 17 Diagrama de caso de uso - Usuario

Ambos actores podrán ver el apartado "Mi Cuenta" y gestionarla, realizando los siguientes casos de uso:

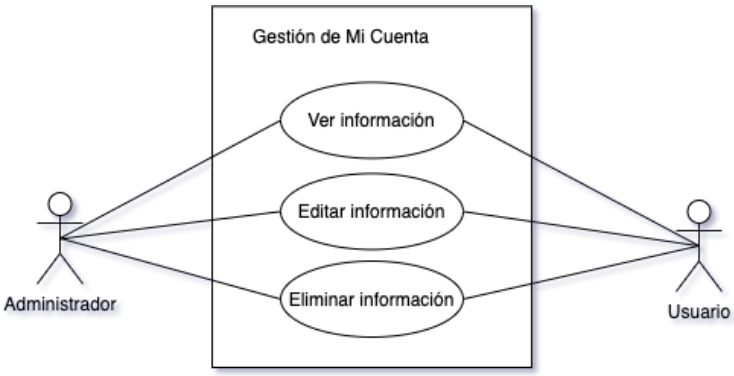


Ilustración 18 Diagrama de caso de uso - Gestión de Mi Cuenta

5.2 Diseño de interfaz web de front-end

En este apartado se detalla la funcionalidad para la plataforma web. Para ello se ha desarrollado un flujo de pantallas, que se encuentra a continuación. Este flujo de pantallas es la base del diseño de la aplicación, la lógica que se va a seguir para desarrollar el código y la arquitectura del proyecto. La plataforma constará de un inicio de sesión inicial, una home o página principal, el *layout* principal constará de un *sidebar* o barra lateral a la izquierda en el que se dispondrá una navegación básica, un *header* o cabecera con un buscador y *breadcrumbs* o barra de navegación. A continuación, se especifican las funcionalidades principales de cada pantalla.

5.2.1 Página principal

En la página principal se podrá visualizar información genérica sobre los tipos de deportes que hay, y el funcionamiento de la propia plataforma.

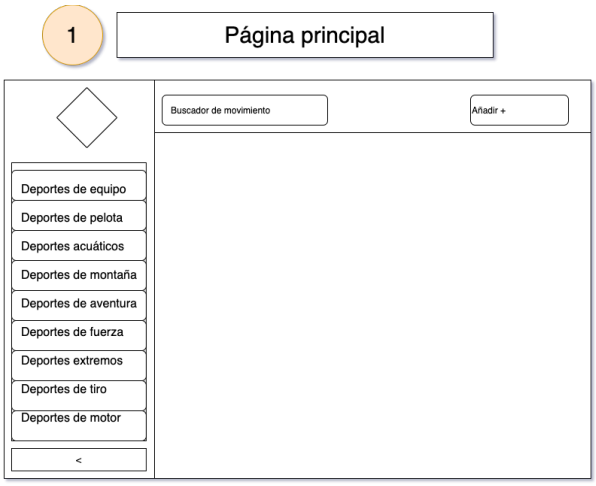


Ilustración 19 Diseño original - Página principal

5.2.2 Selección de tipo de deporte

Al seleccionar un tipo de deporte se podrá visualizar los deportes englobados en este y una breve descripción.

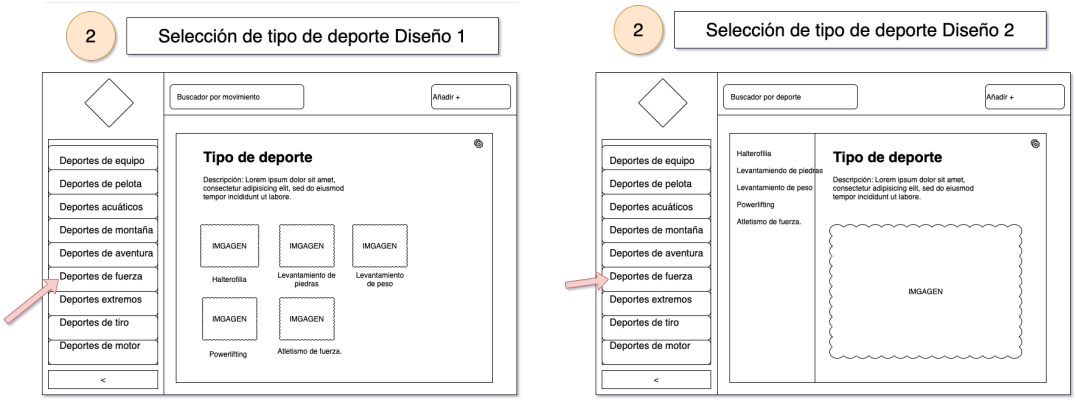


Ilustración 20 Diseño original - Selección de tipo de deporte

5.2.3 Selección de deporte

A la hora de seleccionar un deporte se podrá visualizar una descripción de este y los movimientos que ya tiene configurados.

Ilustración 21 Diseño original - Selección de deporte

5.2.4 Selección de movimiento

Al seleccionar un movimiento se abre el visualizador de movimientos 3D para poder ver el movimiento en acción.

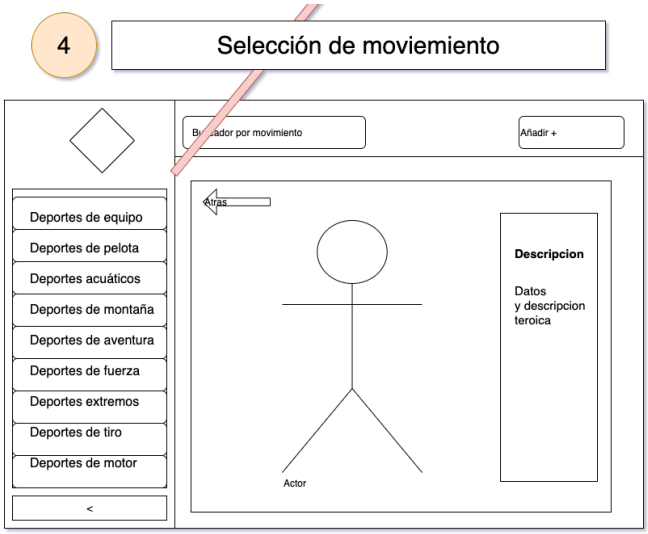


Ilustración 22 Diseño original - Selección de movimiento

5.2.5 Añadir movimiento

Esta opción solo está habilitada para usuarios con rol administrador, y sirve para añadir nuevos movimientos a la base de datos.

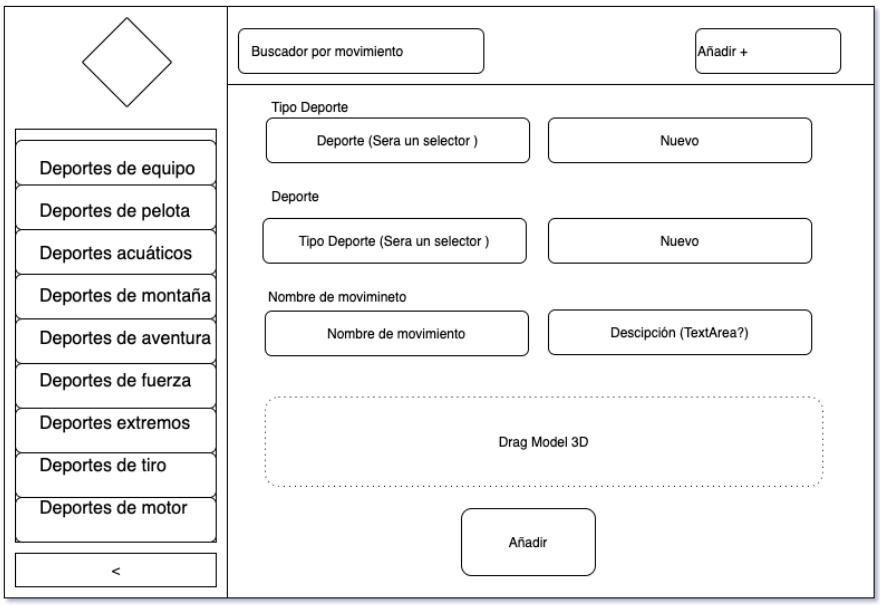


Ilustración 23 Diseño original - Añadir movimiento

5.3 Diseño de back-end

Dentro del back-end y siguiendo el diseño de casos de uso se define y diseña la base de datos que se muestra a continuación:

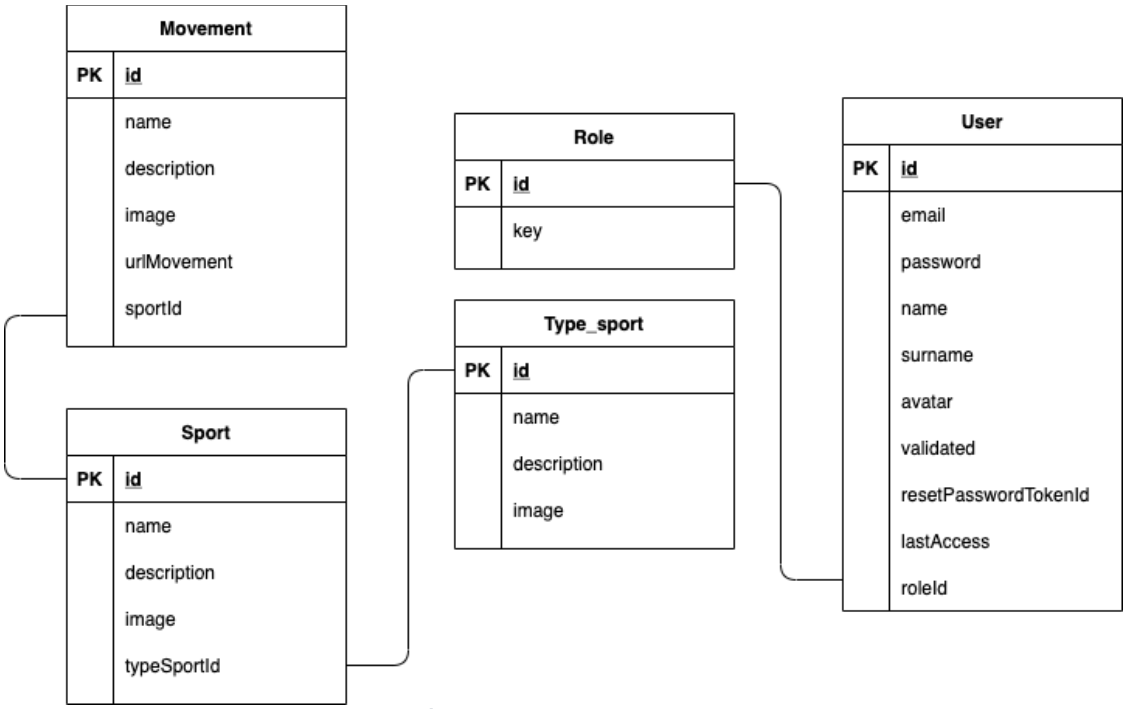


Ilustración 24 Diseño original base de datos

5.4 Investigación y definiciones necesarias en 3D

Antes de seguir hay que decir que llegados a este momento no tenía muy clara la diferencia entre animación, objeto y rigging e investigué estas nomenclaturas, para poder tener todo más claro.

5.4.1 Escena

Una escena 3D es un espacio tridimensional que almacena datos geométricos (a menudo cartesianos), asociados habitualmente con otros tipos de información visual (luces, texturas, etc.), la geometría de una escena se construye a partir de uno o más objetos. Estos objetos pueden variar desde luces para iluminar su escena, formas básicas en 2D y 3D para rellenarla con modelos, esqueletos para animar esos modelos. Una escena suele contar con los siguientes elementos:

- Cámara, establecen el punto de vista que se tendrá en cuenta como referencia para renderizar la escena.
- Modelos u objetos: representaciones matemáticas de las superficies de los objetos en tres dimensiones, habitualmente generados con software especializado. Se almacenan como un conjunto de puntos (vértices) que se unen a usando diversas entidades geométricas (como líneas, triángulos, quads, etc.).
- Luces, puntos a partir de los cuales se generan focos lumínicos.

5.4.2 Modelo u objeto

Cada tipo de objeto (malla, luz, curva, cámara, etc.) se compone de dos partes: un objeto y datos de objeto (a veces abreviado como «ObData»):

<u>Objeto</u>

Contiene información sobre la posición, rotación y tamaño de un elemento en particular.

<u>Datos de objeto</u>

Contiene todo lo demás. Por ejemplo:

- Mallas: almacenar geometría, lista de materiales, grupos de vértices, etc.
- Cámaras: almacene la distancia focal, la profundidad de campo, el tamaño del sensor, etc.

Cada objeto tiene un enlace a sus datos de objeto asociados, y muchos objetos pueden compartir un solo dato de objeto, en el caso de las mallas geométricas, estos datos de objetos es lo que normalmente se llama modelo.

5.4.3 Animación

La animación tiene su origen en el latín "Anima" su significado es "Alma", por lo tanto, animar significa dar alma a un personaje u objeto de forma que parezca que éste cobra vida piense y actúe por sí mismo. Técnicamente la animación 3D se realiza mediante programas o aplicaciones que simulan la visualización tridimensional todo ello se realiza a través de cálculos basados en la proyección de la geometría y espacios tridimensionales sobre pantallas bidimensionales. Estos programas y aplicaciones permiten la creación y manipulación de mallas poligonales que luego son convertidas en los fotogramas que componen cualquier secuencia.

5.4.4 Rigging

La traducción literal de rigging es "aparejo". En el contexto del software de diseño 3D el significado de rigging es el proceso por el cual creamos, a partir de un modelo o escultura digital tridimensional (3D), la estructura necesaria para poder deformar y animar personajes. De la misma forma que haríamos con una marioneta, se crea una estructura de huesos y controles (rigs) dentro de una geometría para que los animadores puedan animar el cuerpo mediante dichos rigs.

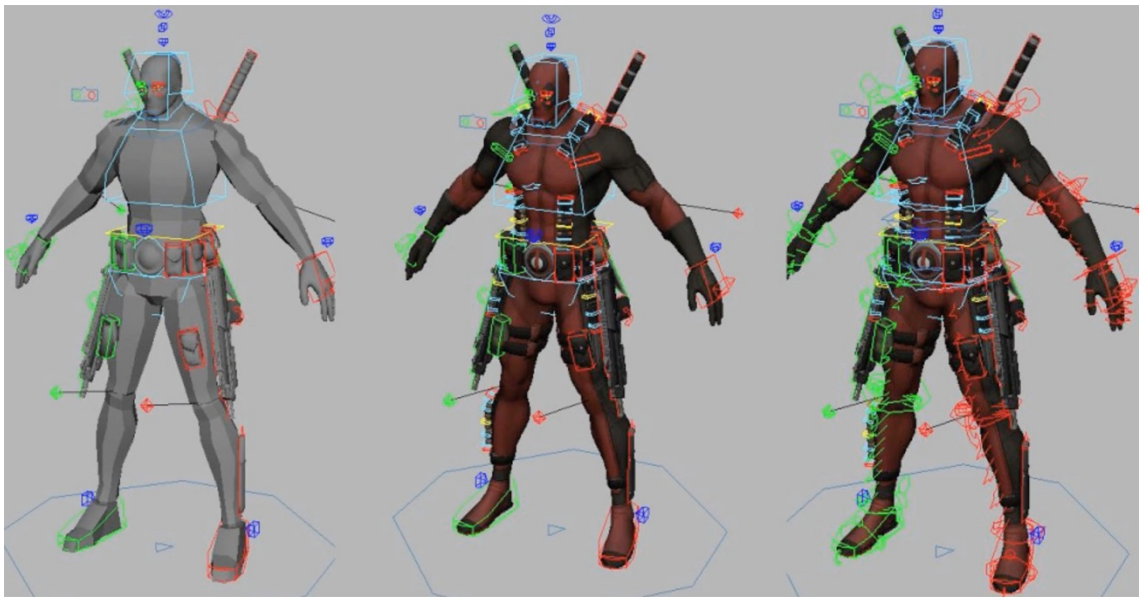


Ilustración 25 Rig model [35]

5.5 Análisis del conversor

A la hora de obtener información sobre los modelos 3D, nos decidimos por el modelo FBX para visualizarlo a través de la aplicación web con la librería Three.js. Por ello la conversión de los archivos será al formato FBX.

5.5.1 Conversores

En esta fase del proyecto se destina mucho tiempo a la investigación sobre conversores y el propio formato mdx. No se encuentra ningún conversor válido por lo que nos decidimos a realizar un conversor propio, este será de formato mdx a FBX, y la mejor elección es hacerlo en Unity ya que el tutor posee mucha experiencia con el motor, y puede ayudarme.

Como explicamos en el apartado de antecedentes tenemos exportaciones que provienen de diferente software. A la hora de hacer el conversor se decide que se hará a formato FBX ya que es el más sencillo para tratar con la librería Three.js. Por lo tanto, tendremos la exportación que viene del sistema BTS SMART-D de la marca BTS Bioengineering y el archivo FBX que se descarga del sistema Wearnotch.

5.5.2 Exportación de BTS SMART-D, análisis de formato mdx

Como bien se ha dicho anteriormente el formato mdx sigue una estructura XML, al abrirlo en el navegador podemos visualizar la siguiente estructura:


```xml
<?xml version="1.0"?>

<emxDataFile format="0.1" sourceApp="Viper">
  <trial
    label="0017~ab~Lateral Bending.tdf"
    description="SMART acquisition"
    date="1-1-2000"
    time="0:0:0"
  >
    <stream
      label=""
      description=""
      startTime="0"
      frequency="100"
      nFrames="1726"
    >
      <track
        label="AcromionDn"
        description=""
        nItems="-1"
        nSegs="1"
        nPoints="1726"
        coords="X Y Z"
        scaleFactor="10000"
        data="S 0 2257 10093 13456 2257 10093 13457 2257 10093 13457 2256 10093 13457 2257 10093 13457 2257 10093 13457 2256 10093 13455 2255
        10093 13453 2256 10093 13454 2257 10093 13456 2257 10093 13456 2257 10093 13456 2257 10093 13456 2257 10094 13456 2257 10094 13456 2257
        10094 13456 2257 10094 13456 2257 10094 13456 2257 10094 13456 2257 10094 13456 2257 10094 13456 2257 10093 13456 2257 10093 13456 2257
      />
      <track
        label="AcromionSn"
        description=""
        nItems="-1"
        nSegs="1"
        nPoints="1726"
        coords="X Y Z"
        scaleFactor="10000"
```


Ilustración 26 Estructura formato mdx

Al revisar este formato deducimos que los *track* pertenecen a cada *bone* del esqueleto, donde *label* hace referencia a la jerarquía vista con anterioridad en el apartado de antecedentes,

en el data parece que se almacenan "S 0 x y z x z y z …" Será la posición de este *bone* en el espacio en cada *frame*.

En la implementación encontraremos una solución a este análisis, con un conversor que ponga las coordenadas de estos *bones* en una estructura en formato FBX.

5.5.3 Exportación de Wearnotch

WearNotch nos proporciona archivos directamente en formato FBX, esto puede parecer trabajo ya hecho, pero al abrir estas exportaciones en Blender [36], que es un programa informático multiplataforma, dedicado al modelado 3D entre otras cosas, la animación está, pero no hay un objeto en ella, es decir, tenemos el rig pero no el body. Esto nos ocasiona un gran problema ya que para poder visualizar este movimiento a través de Three.js es necesario tanto el modelo como la animación. Y este modelo debe tener un *rig* bien hecho con una pose adecuada y una estructura afín.

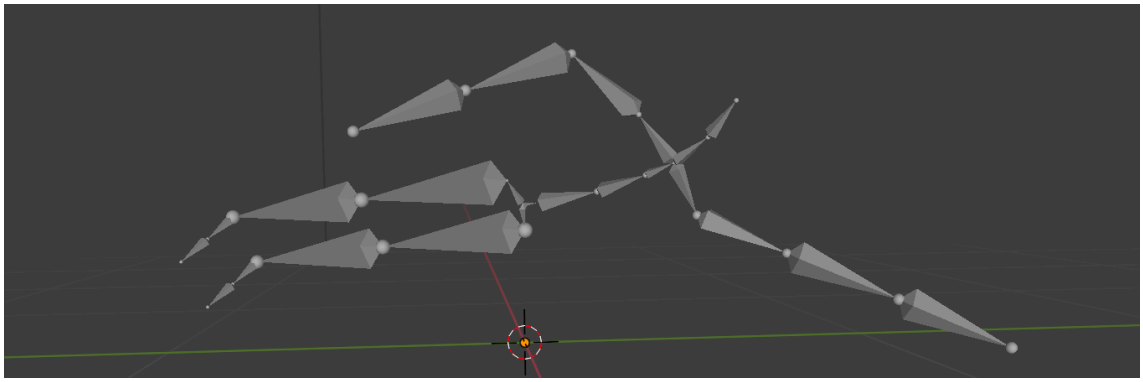


Ilustración 27 Exportación FBX de Wearnotch

En la implementación podremos ver como se resuelve este problema.

6 Implementación

Después de analizar el problema y diseñar los diferentes componentes necesarios para desarrollar la solución al proyecto se comienza con la implementación de este. Siguiendo el orden de planificación estipulado.

6.1 Diseño y arquitectura back-end

Comenzamos con la implementación del diseño y arquitectura del back-end, ya que esto nos permitirá tener una base de datos sólida y con la que poder trabajar desde el frontal.
Al realizar la estructura con NestJs hubo un primer momento de búsqueda de información, ya que hasta este momento no había realizado ningún proyecto con esa tecnología.

Si desgranamos un poco este framework se puede ver que está basado en Node.js y Express. Además, permite construir un back-end en TypeScript con el patrón MVC, modelo vista controlador.

Destacando algunas de sus características principales se puede decir que:

1. Cuenta con lenguaje tipado.
2. Separación por módulos.
3. Generación automática de entidades a partir de una base de datos.
4. ORM, Object Relational Mapper.
5. Construcción de endpoints a través de decoradores.
6. Inyección de dependencias.
7. Soporte a Swagger.

Lo que más me gusta de este framework como desarrolladora con experiencia front-end en Angular es que NestJs está influenciado en gran medida por Angular, aprovechando muchos de sus elementos y conceptos como son:

- Módulos: estos módulos agrupan un conjunto de artefactos Angular, como son componentes, directivas, pipes y servicios que forman parte de ese mismo módulo, es decir, que representa una agrupación lógica de scripts creando así un área cuasi independiente al que podemos llamar una pequeña área funcional de nuestra aplicación. Estos módulos contienen parte específica del desarrollo que encapsulada puede migrarse a otro proyecto y reutilizarse. Por ejemplo, todos los scripts necesarios para realizar las autorizaciones irían dentro de un mismo módulo.

- Inyección de dependencias, es un patrón de diseño en el que una clase requiere instancias de una o más clases y, en vez de generarlas dentro de su propio constructor, las recibe ya instanciadas por un mecanismo externo.

En este sentido, también cuenta con su propia CLI (command-line interface), es un método que permite a los usuarios dar instrucciones al programa informático por medio de una línea de texto simple, para generar y facilitarnos las tareas de creación de todos estos elementos.

A la hora de utilizar NestJs lo primero que me sorprendió fue el uso de la biblioteca TypeORM, ya que facilita en gran medida la programación de entidades y su relación y a su vez las características y beneficios que aporta a la hora de programar me parecían muy útiles.

TypeORM es un ORM, Object Relational Mapper, biblioteca que se ejecuta en Node.js y está escrita en TypeScript. TypeScript es una modificación del lenguaje JavaScript, es un lenguaje 'compilado'. No se interpreta en tiempo de ejecución. El compilador TypeScript toma archivos TypeScript (.ts) y los compila en archivos JavaScript (.js), a este proceso se le llama transpilación.

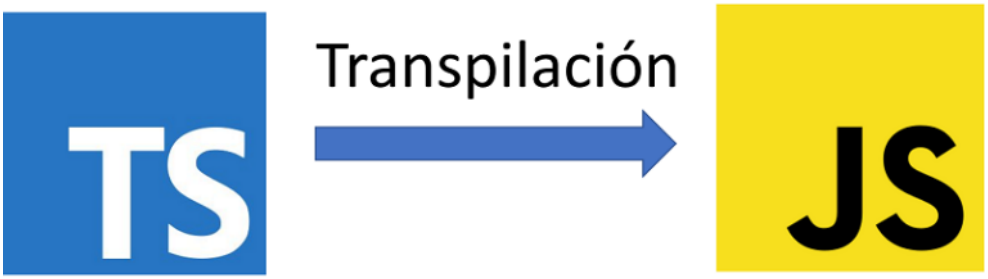


Ilustración 28 Gráfico de transpilación

TypeORM admite conexión con diferentes bases de datos, pero en este proyecto se ha seleccionado MySQL. La funcionalidad TypeORM son conceptos específicos de RDBMS, sistema de gestión de bases de datos relacionales.

6.1.1 Características de TypeORM

TypeORM posee diferentes características que facilitan el diseño entidad-relación entre tablas de la base de datos, puedes:

1. Crear automáticamente esquemas de tablas base de datos basados en sus modelos predefinidos en el proyecto NestJs.
2. Puedes insertar, actualizar y eliminar fácilmente objetos en la base de datos.

3. Se puede crear un mapeo relacional entre tablas, del tipo:
    a. one-to-one, un objeto de la entidad dada se relaciona con un solo objeto de la entidad de destino y viceversa.
    b. many-to-one, el objeto múltiple de la entidad dada se relaciona con un objeto de la entidad de destino.
    c. one-to-many, el objeto de la entidad dada se relaciona con múltiples objetos de la entidad de destino.
    d. many-to-many, el objeto múltiple de la entidad dada se relaciona con el objeto múltiple de la entidad de destino.
4. Proporciona comandos CLI (command-line interface) simples.

6.1.2 Beneficios de TypeORM

TypeORM es un framework ORM fácil de utilizar con codificación simple. Tiene los siguientes beneficios:

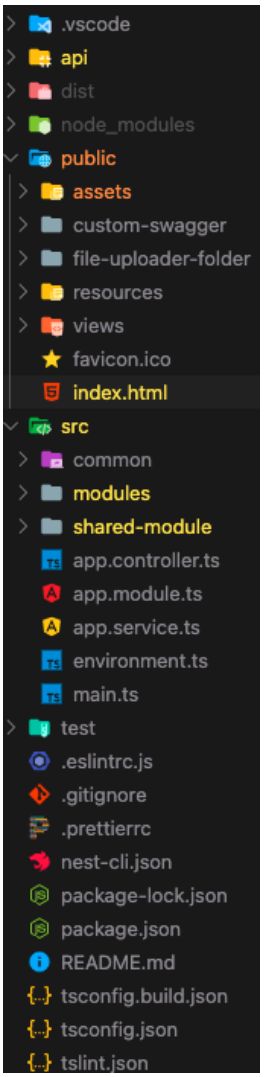
1. Aplicaciones de alta calidad y poco acopladas.
2. Aplicaciones escalables.
3. Es fácilmente integrable con otros módulos.
4. Se adapta perfectamente a cualquier arquitectura, desde aplicaciones pequeñas hasta empresas.

De estos beneficios lo que realmente nos interesa es la aplicación de alta calidad y poco acoplada, ya que nos da flexibilidad a la hora de hacer cambios, la posibilidad que sea escalable para el crecimiento del proyecto en el futuro y la facilidad que aporta a la ahora de integrarlos con otros módulos. También la adaptación a cualquier arquitectura brinda la facilidad de poder usarlo en cualquier aplicación.

6.1.3 Estructura proyecto

Al comenzar con este proyecto, se crea un proyecto NestJs llamado "nestjs-3dmovement-to-biomechanical" conectado a Github y que gestiono mediante Sourcetree, con la siguiente arquitectura:

- node_modules: módulos npm guardados localmente.
- src: contiene el código fuente de su aplicación en lenguaje TypeScript.
- src/main.ts: su punto de entrada a la aplicación.
- src/environment.ts: archivo de configuración principal de su aplicación. Contiene detalles de configuración de la base de datos y configuración de entidades.
- src/common: esta carpeta almacena los scripts referentes a scripts comunes a todos los módulos y que se pueden utilizar en cada uno de ellos.
- src/modules: esta carpeta contiene diferentes módulos definidos en el proyecto, y los crud, que explicaremos más tarde.
- src/shared-module: esta carpeta contiene módulos transversales e independientes.
- package.json: contiene las dependencias del módulo de nodo.
- package-lock.json: archivo generado automáticamente y relacionado con package.json.
- tsconfig.json: contiene opciones de compilador específicas de TypeScript.



6.1.4 CRUD

CRUD (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. Dentro de la carpeta "shared-module" se ubica el módulo "crud" este módulo lo que nos permite es optimizar y reutilizar el código, este está conectado a Swagger. Digamos que este CRUD nos sirve como plantilla para las llamadas básicas de crear, leer, actualizar y borrar de nuestro back-end.

En este CRUD concreto tendremos las siguientes llamadas:

GET	/api/v1/nombre_entidad
	Obtener todas las entidades paginadas

POST	/api/v1/nombre_entidad
	Crea entidad
GET	/api/v1/nombre_entidad/{id}
	Obtiene entidad por id
DELELTE	/api/v1/nombre_entidad/{id}
	Borra entidad por id
PUT	/api/v1/nombre_entidad/{id}
	Actualiza entidad por id

Tabla 2 Llamadas CRUD

Cada controlador que extienda este CRUD podrá hacer uso de estas llamadas pasando su propia entidad.

Dentro de este módulo tendremos el archivo CrudEntity que tiene las propiedades básicas para cada entidad, las entidades que extiendan de esta clase heredaran sus propiedades. Por otro lado, tenemos el CrudEntityController, que tendrá las llamadas definidas en la *Tabla 2,* y el CrudEntityService, que contiene funciones ya definidas para realizar las consultas a la base de datos.

6.1.5 Swagger

Swagger es un conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful. Actualmente este software se llama OpenAPI Specification (OAS), define una interfaz estándar, independiente del idioma, para las API RESTful que permite visualizar y comprender las capacidades del servicio sin acceso al código fuente. Esta interfaz nos permite interactuar directamente con nuestra base de datos, de forma que todos los endpoints que generemos en la aplicación quedarán debidamente documentados en la URL:

url:3000/api/v1/docs

Se puede visualizar la vista de swagger en el Anexo VII – Swagger.

Para poder saber qué propiedades son las adecuadas a la hora de programar, swagger tiene su propia anotación a través de decoradores para así poder identificar cada propiedad. Por ejemplo, al definir una entidad, se añaden ciertos decoradores para identificar de qué tipo va a ser cada columna:

```
@Entity()
export class TypeSport extends CrudEntity {
  @ApiModelProperty({ type: String })
  @Column({ type: String })
  name: string;

  @ApiModelProperty({ type: String })
  @Column({ type: String, default: '' })
  description: string;
```

Decorador @ApiModelProperty

Decorador @ApiModelProperty

Ilustración 29 Ejemplo decoradores swagger

Algunos de los decoradores que se han utilizado son los siguientes:

- ApiModelProperty, define como campo de descripción, permite configurar varias propiedades del objeto de esquema.
- ApiOperation, define la función del método de descripción.
- ApiResponse, clase de respuesta.
- ApiImplicitBody, para establecer explícitamente la definición del cuerpo.

Para más información se puede visualizar la documentación de OpenAPI [37].

6.1.6 Conexiones

La configuración disponible del back-end está definida en el archivo environment.ts mediante este archivo se definen las opciones de tipo, host, nombre de usuario, contraseña, base de datos y puerto que están relacionadas con la configuración de la base de datos MySQL.

6.2 Base de datos

Al tener el proyecto "nestjs-3dmovement-to-biomechanical" y desplegarlo con una conexión estable a la base de datos, que en este caso está creada en MySQL usando la herramienta visual de diseño de bases de datos MySQL Workbench [38], se autogenerarán las tablas en la base de datos y el resultado es el siguiente:

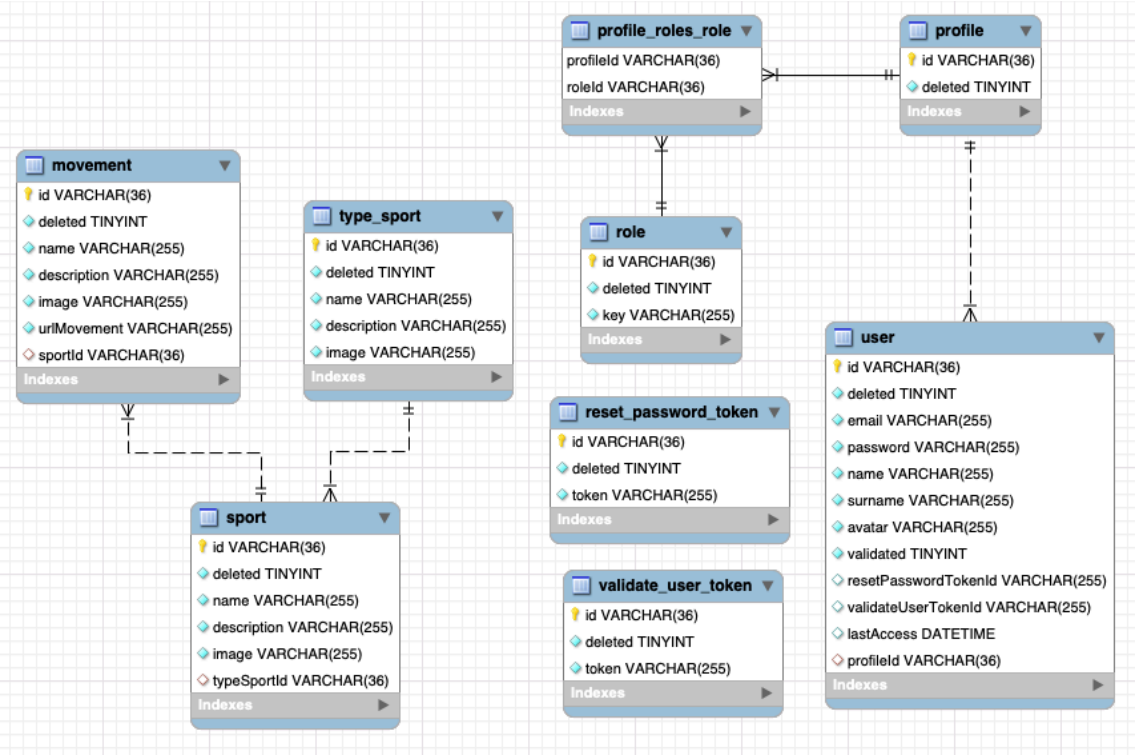


Ilustración 30 Diagrama de Base de datos MySQL Workbench

6.3 Desarrollo – Auth

Antes de empezar hay que saber que hay dos partes diferentes en cuanto a la seguridad de una aplicación web: autenticación y autorización.

El desarrollo de Auth se hace apoyándose en la autentificación JWT y la autorización mediante roles.

6.3.1 Autenticación

La autenticación es el proceso mediante el cual verificas que un usuario es quien dice ser en tu aplicación. En el proyecto esto lo haremos mediante el log-in o inicio de sesión.

6.3.2 Autorización

La autorización permite configurar las áreas a las que tendrán acceso los usuarios autenticados.

6.3.3 ¿Qué es JWT?

JWT (JSON Web Token) es un estándar enmarcado dentro del RFC 7519. En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de privilegios. Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro del payload o cuerpo de un mensaje que va firmado digitalmente.

Este objeto de tipo objeto JSON se llama token y son una cadena de texto que tiene tres partes codificadas en Base64, cada una de ellas separadas por un punto. Cada una de estas partes se la siguiente:

- Header: encabezado dónde se indica, al menos, el algoritmo y el tipo de token.
- Payload: donde aparecen los datos de usuario y privilegios, así como toda la información que queramos añadir, todos los datos que creamos convenientes.
- Signature: una firma que nos permite verificar si el token es válido.

Los Token JWT tienen el siguiente ciclo de vida:

Ilustración 31 Ciclo de vida de un token JWT [39]

El proyecto NestJs tiene instalados el paquete "jsonwebtoken" para poder realizar esta verificación.

6.4 Desarrollo – Roles

Los roles son definidos dentro del módulo Auth, son en sí una entidad como tal, con la siguiente estructura:

```
id: string
deleted: boolean
key: string
```

6.5 Desarrollo – Users

A la hora de desarrollar los usuarios se crea un módulo como tal llamado "user". Los usuarios serán una entidad en sí misma y extienden de CrudEntity, y serán una tabla en nuestra base de datos, con las siguientes propiedades:

```
id: string
deleted: boolean
email: string
password: string
name: string
surname: string
avatar: string
profile: Profile
validated: Boolean
resetPasswordTokenId: string
validatedUserTokenId: string
lastAccess: Date
```

Se puede apreciar que cada entidad usuario está relacionada con un *profile*, el *profile* se define de la siguiente manera:

```
id: string
deleted: boolean
rol: Role[]
```

6.6 Desarrollo - Tipos de deporte, deportes y movimientos

Se puede decir que el desarrollo de estos tres módulos es muy similar, cada uno de ellos extiende del CrudEntityController y sus entidades extienden también de CrudEntiy. Siguen el diseño original de la base de datos.

A la hora de definir la relación entre las tablas se hace de la siguiente manera:

```
@Entity()
export class TypeSport extends CrudEntity {
  @ApiModelProperty({ type: String })
  @Column({ type: String })
  name: string;

  @ApiModelProperty({ type: String })
  @Column({ type: String, default: '' })
  description: string;

  @ApiModelProperty()
  @Column({ type: String, default: '' })
  image: string;

  // @ApiModelProperty({ type: () => [Sport] })
  @OneToMany(
    type => Sport,
    sport => sport.typeSport,
    { eager: true },
  )
  @ValidateNested({ each: true })
  @Type(() => Sport)
  sports: Sport[];
}
```

```
@Entity()
export class Sport extends CrudEntity {
  @ApiModelProperty({ type: String })
  @IsString()
  @Column({ type: String })
  name: string;

  @ApiModelProperty({ type: String })
  @IsString()
  @Column({ type: String })
  description: string;

  @ApiModelProperty({ type: String })
  @IsString()
  @Column({ type: String, default: '' })
  image: string;

  @ApiModelProperty({ type: () => TypeSport })
  @ManyToOne(type => TypeSport)
  @ValidateNested({ each: true })
  @Type(() => TypeSport)
  typeSport: TypeSport;

  // @ApiModelProperty({ type: () => [Movement] })
  @OneToMany(
    type => Movement,
    movements => movements.sport,
    { eager: true },
  )
  @ValidateNested({ each: true })
  @Type(() => Movement)
  movements: Movement[];
}
```

```
@Entity()
export class Movement extends CrudEntity {
  @ApiModelProperty({ type: String })
  @Column({ type: String })
  name: string;

  @ApiModelProperty({ type: String })
  @Column({ type: String })
  description: string;

  @ApiModelProperty({ type: String })
  @IsString()
  @Column({ type: String, default: '' })
  image: string;

  @ApiModelProperty({ type: String })
  @IsString()
  @Column({ type: String, default: '' })
  urlMovement: string;

  @ApiModelProperty({ type: () => Sport })
  @ManyToOne(type => Sport)
  @ValidateNested({ each: true })
  @Type(() => Sport)
  sport: Sport;
}
```

Ilustración 32 Entidades de tipo de deporte, deporte y movimiento

De puede apreciar en los decoradores de las propiedades la relación, one-to-many y many-to-one entre las entidades.

6.7 Desarrollo file-upload

El módulo file-upload es muy importante ya que es el módulo clave para poder subir contenido a la base de datos, es módulo está ubicado en la carpeta shared-module, y sirve para poder subir archivos de diferentes formatos. En el FileUploaderController hay definidas diferentes llamadas, que podemos ver en la siguiente captura de swagger:

file-uploader

POST	/api/v1/file-uploader/multiple-files	Upload multiple files
POST	/api/v1/file-uploader/avatar	Upload image
POST	/api/v1/file-uploader/object	Upload Object

Ilustración 33 Vista de endpoints del módulo file-uploader en swagger

A la hora de subir archivos se definen y filtran a través del tipo MIME (extensión multipropósito de Correo de Internet), es una forma estandarizada de indicar la naturaleza y el formato de un documento, archivo o conjunto de datos. Siguiendo la estructura general de MIME tipo/subtipo, se implementa la lógica de subir imágenes para que solo admita archivos en formato: jpeg, jpg o png.

A la hora de subir objetos solo admite objetos de tipo application/octet-stream, este es el valor predeterminado para un archivo binario. Como realmente significa un archivo binario desconocido, en este caso admite FBX.

6.8 Conversor

A la hora de desarrollar el conversor en Unity, se plantea realizar un editor que nos permita seleccionar el archivo FBX que queremos convertir y la ruta donde queremos exportarlo. El flujo que hay que seguir para convertir un archivo y exportarlo es el siguiente:

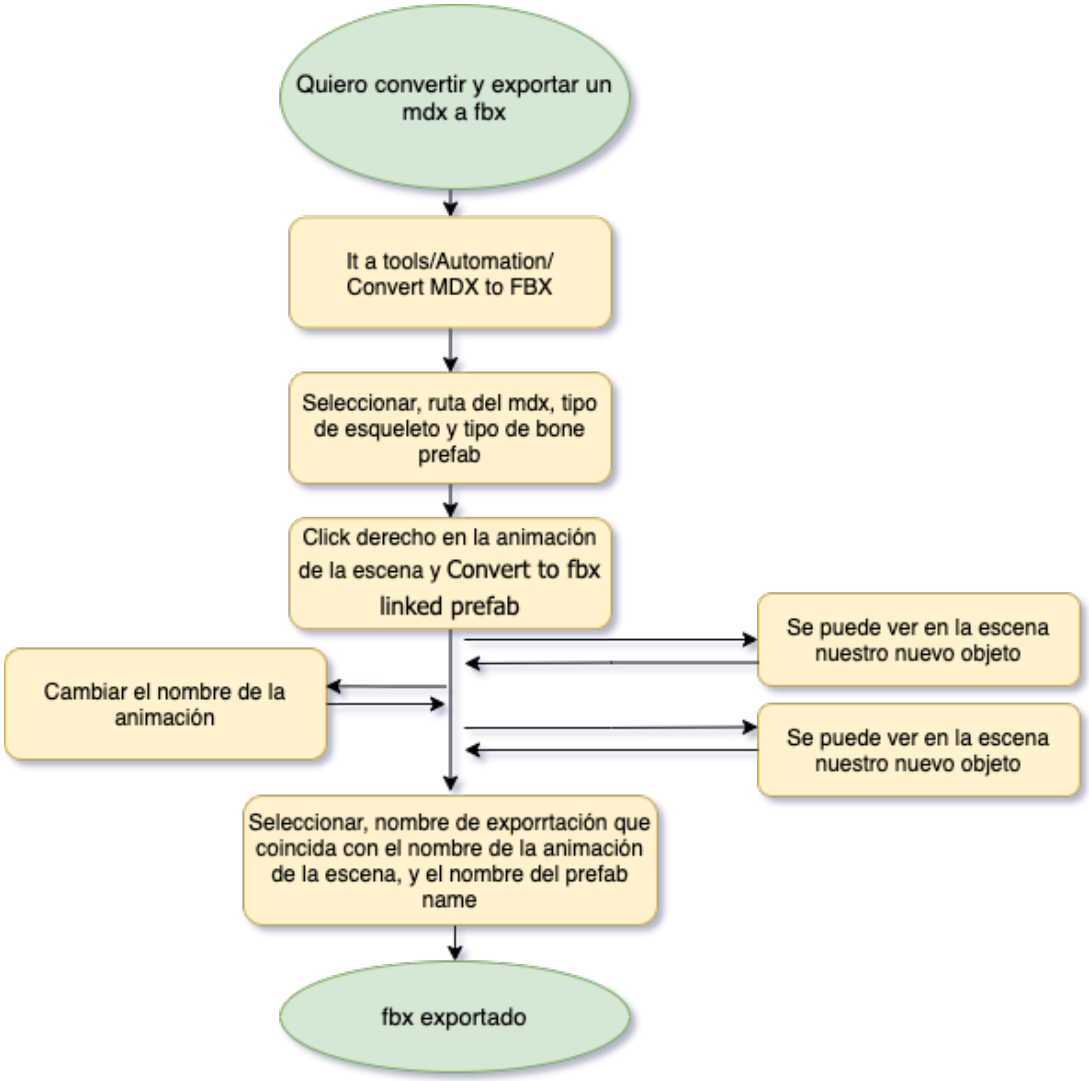


Ilustración 34 Diagrama de flujo conversión de mdx a FBX

Se define un esqueleto que sigue la jerarquía facilitada por el personal del laboratorio de la Universidad San Jorge, que a su vez, son los tracks de los archivos mdx, con todos sus nombres de referencia. La jerarquía se establece en dos versiones, el upperbody, que corresponde a las capturas de movimientos del tren superior del cuerpo y el lowerbody que corresponde a los movimientos capturados del tren inferior del cuerpo. Cada uno de estos puntos son llamados *bone*, y son instancias del prefab seleccionado en este caso una esfera gris.

Este prefab puede ser mofificado en cualquier momento, o generar otros que puedan servirnos, y seleccionarlos al convertir.

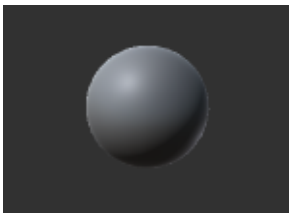


Ilustración 35 Bone Prefab

Estos esqueletos definidos serán la base estructural de nuestro FBX, pero este ha de ser animado, por lo tanto, hay que mover cada *bone* en el tiempo. Esta información está ya disponible en los archivos mdx, ya que en cada *track* se dispone de la posición relativa en el mundo de cada *bone*. La lógica de este proceso está implementada dentro del script 'AnimBone' y la estructura predefinida de los esqueletos en el script 'PredefinedSkeleton'.

Para facilitar el proceso de conversión se crean editores todos ellos accesibles desde la opción 'Tools', los editores son los siguientes:

- MDX Converter, sirve para convertir el archivo mdx a un objeto dentro de la escena.
- Skeleton to Animation, convierte el objeto de la escena y crea una animación.
- Automation > Convert MDX to FBX, convierte el MDX a un objeto y este a una animación en la escena.

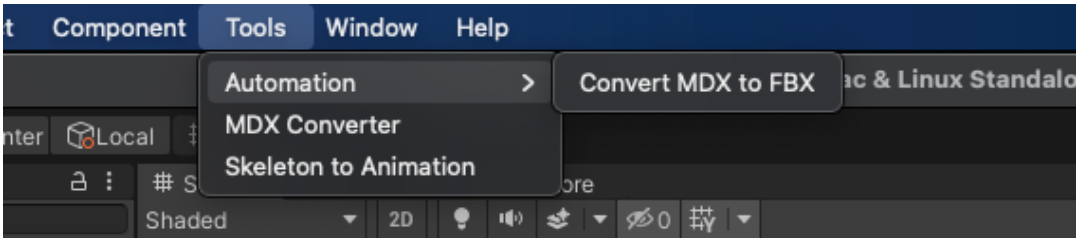


Ilustración 36 Editores en Tools

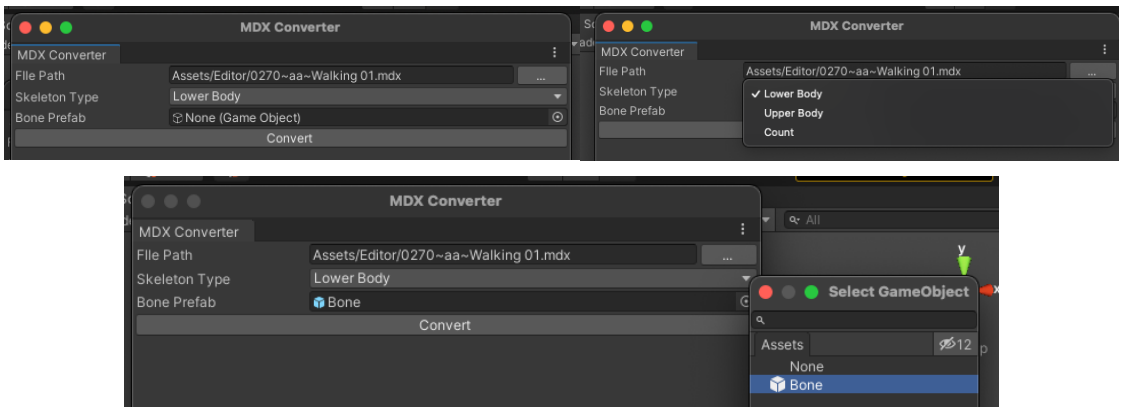


Ilustración 37 MDX Converter

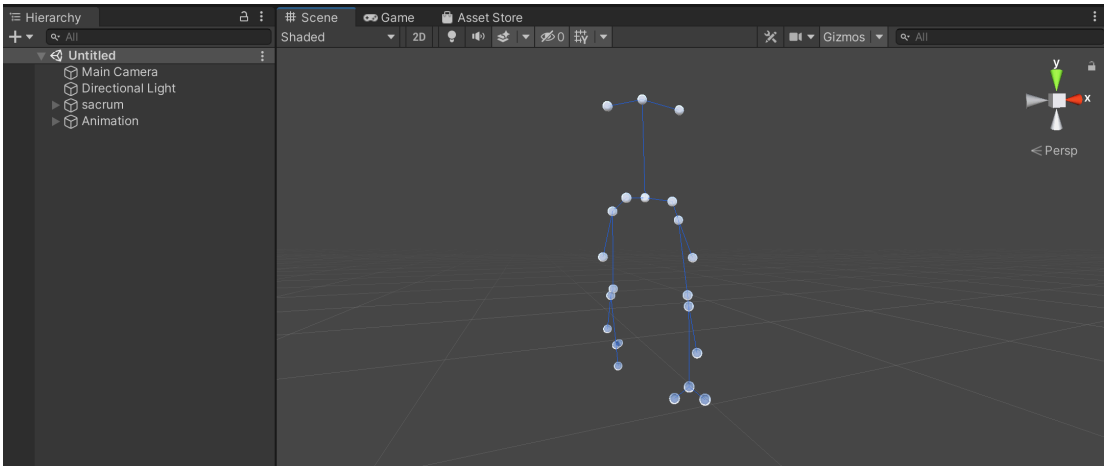


Ilustración 38 Vista de la escena de unity con el objeto y la animación

Una vez están el objeto animado en la escena, hay que exportar a FBX, para ello de forma opcional se puede cambiar el nombre a la Animation de la escena, y después click derecho y seleccionamos 'Convert to FBX linked prefab', rellenamos el editor con la condición de que el 'Export name' y el 'Prefab name' tienen que coincidir con el nombre de la animación en la escena. Al convertir se habrá creado un FBX en Assets/Models.

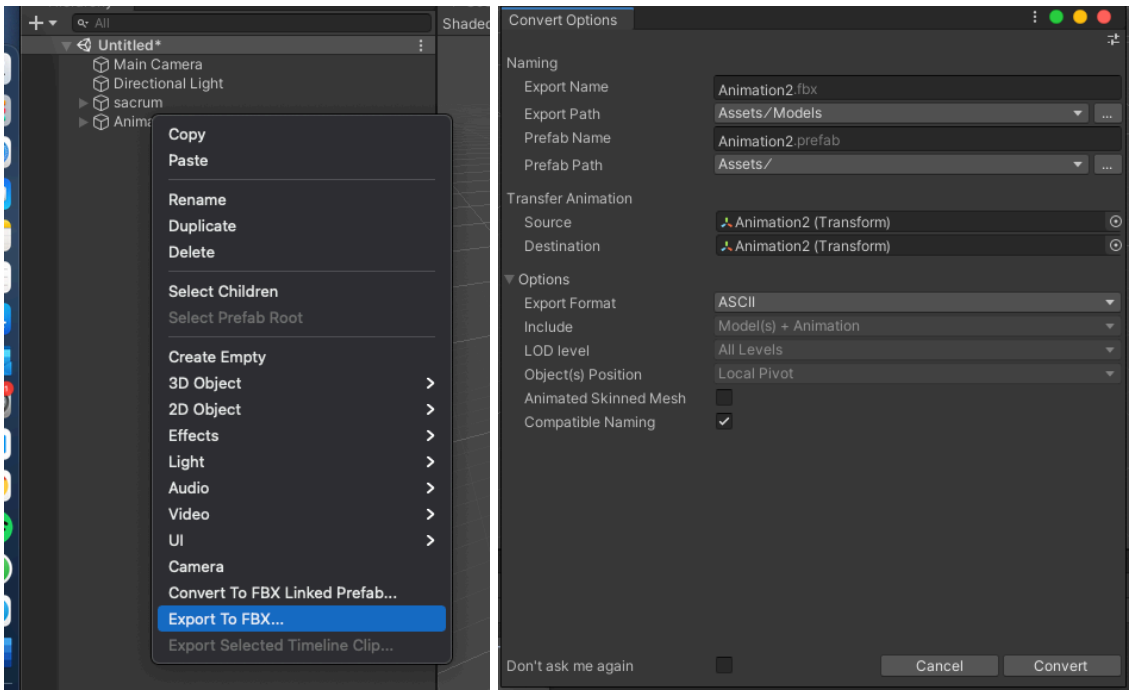
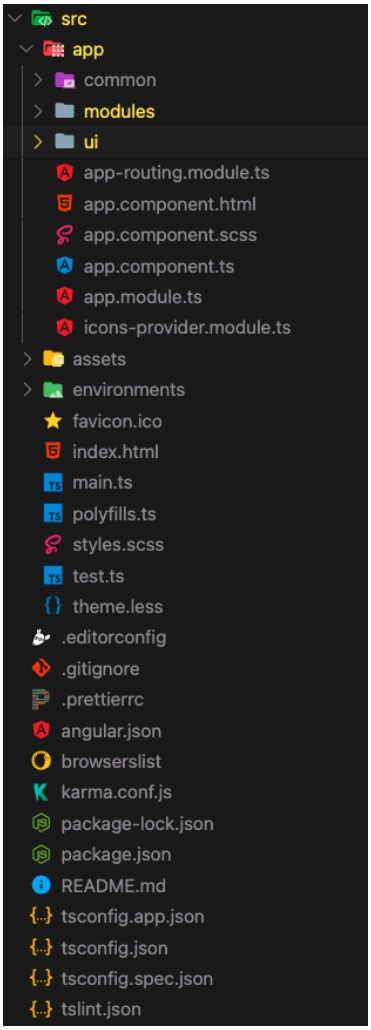


Ilustración 39 Editor Convert to FBX linked prefab

6.9 Diseño y arquitectura front-end

Llegados a este punto se comienza con la implementación del diseño y arquitectura del front-end, este es un proyecto realizado en Angular 8 al que se le pone el nombre "angular-front-dashboard-movement-to-biomechanic", la estructura de este proyecto tiene una arquitectura básica.

- node_modules: es la carpeta que contiene todas las dependencias de nuestro proyecto.
- src: es el directorio donde se desarrolla el código de la aplicación, y se distribuyen los módulos.
- src/app, aquí se ubica toda la implementación de los componentes principales, junto a su template, html y archivos de estilos scss.
- src/app/common, en la carpeta common se encuentras las clases, componentes, guards, interceptores, pipes y servicios comunes en toda la aplicación.
- src/app/modules, en esta carpeta se encuentran definidos los módulos de la aplicación.
- src/app/ui, se define el layout y la estructura principal de la vista de la aplicación.
- assets, contendrá todos los asset y archivos adicionales para hacer que el proyecto funcione.



- enviroments, donde se encuentra las configuraciones y variables de entorno para poner el proyecto tanto en desarrollo como en producción.
- favicon.ico, Es el archivo del icono del proyecto.
- index.html, es el archivo de la página principal del proyecto.
- main.ts, es el archivo Type Script inicial del proyecto donde podrás configurar todas las configuraciones globales del proyecto.
- editorconfig: es la configuración de nuestro editor de código.
- .gitignore: son las carpetas o archivos que debe ignorar el git cuando lo añadamos al repositorio.
- angularjson: contiene la configuración de Angular. Además, incluye rutas, versiones, etc.
- package.json: es la configuración de nuestra aplicación. Contiene el nombre de la app, las dependencias necesarias para su correcta ejecución, entre otras cosas.
- REARME.md: aquí podemos añadir información sobre la aplicación. Este archivo es leído por GIT y los muestra en el repositorio.
- tsconfig.json: contiene la configuración TypeScript.
- tslint.json: se utiliza para que el código sea sostenible y se mantenga.

Algo a destacar y mencionar de esta estructura es el archivo src/app/app-routing.module.ts en este archivo se almacenan las rutas y las migas de pan de toda la aplicación:

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { NotFoundComponent } from './ui/views/not-found/not-found.component';

const routes: Routes = [
  { path: 'auth', loadChildren: () => import('./modules/auth/auth.module').then((m) => m.AuthModule) },
  {
    path: 'home',
    loadChildren: () => import('./modules/home/home.module').then((m) => m.HomeModule),
    data: { breadcrumb: 'Home' },
  },
  {
    path: 'visor',
    loadChildren: () => import('./modules/visor/visor.module').then((m) => m.VisorModule),
    data: { breadcrumb: 'Tipos de deporte' },
  },
  {
    path: 'type-sports',
    loadChildren: () => import('./modules/type-sports/type-sports.module').then((m) => m.TypeSportsModule),
    data: { breadcrumb: 'Gestor de Tipo de Deportes' },
  },
  {
    path: 'sports',
    loadChildren: () => import('./modules/sports/sports.module').then((m) => m.SportsModule),
    data: { breadcrumb: 'Gestor de Deportes' },
  },
  {
    path: 'movements',
    loadChildren: () => import('./modules/movements/movements.module').then((m) => m.MovementsModule),
    data: { breadcrumb: 'Gestor de Movimientos' },
  },
  {
    path: 'users',
    loadChildren: () => import('./modules/users/users.module').then((m) => m.UsersModule),
    data: { breadcrumb: 'Gestor de Usuarios' },
  },
  { path: '404', component: NotFoundComponent },
  { path: '', pathMatch: 'full', redirectTo: '/home' },
  { path: '**', redirectTo: '/404' },
];

You, 9 months ago | 1 author (You)
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

Ilustración 40 Código app-routing.module.ts

Cada *path* corresponde a la ruta del navegador, según la ruta se cargará un módulo u otro, van en orden de prioridad descendente, la más prioritaria es la más alta.

6.10 Desarrollo – Auth

En el frontal tenemos diferentes módulos el primero a desarrollar es el de Auth, para poder tener establecida una conexión con el back-end y la base de datos, y poder dotar a la aplicación de un log-in y control de usuarios desde el principio. Este módulo está compuesto por dos componentes el log-in y el sign-up, el log-in sirve para el inicio de sesión y el sign-up para el registro de usuarios. Para poder hacer el inicio de sesión completo se crea la clase token.ts.

6.11 Desarrollo - Implementar store

Una vez está realizado el log-in y en funcionamiento se implementa la store con la librería NGXS una librería de manejo de estado y patrón para Angular, una biblioteca de administración de estado para aplicaciones Angular inspiradas en Redux, es una librería JavaScript de código abierto para el manejo del estado de las aplicaciones. Al usar esta biblioteca, podemos mantener el estado actual de la aplicación en un solo lugar la 'store'.

Tener la información en el cliente también nos permite usarlo como un caché local de datos, cosa que las aplicaciones de Angular no tienen. Nuestra aplicación puede procesar elementos instantáneamente si ya existen en el estado almacenado sin tener que reenviar una solicitud a la API.

Ngxs está basada en el patrón CQRS, Command Query Responsability Segregation, es un estilo arquitectónico en el que tenemos dos subsistemas diferenciados, uno responsable de los comandos, y otro responsable de las consultas. Este patrón utilizado en librerías como Redux, y estos son los módulos principales que la componen:

- Store: Corresponde a un contenedor global que engloba las acciones, estado y selectores.
- Actions: Corresponden a clases que describen una acción a realizar. Poseen metadatos asociados, los metadatos entregan una descripción que utilizamos para saber que está ocurriendo con el estado.
- State: Corresponden a clases que definen el estado de nuestra aplicación. Podemos tener uno o varios.
- Selects: Selectores que toman una pequeña porción del estado para ser utilizado.

Para más información [40]

El funcionamiento principal de esta biblioteca es el uso de la store y de las actions. Todo parte de la acción del usuario, se almacena dicho estado en el store, del store pasa al componente deseado y muestra el resultado en la vista y/o ejecuta otra acción. Se puede ver el funcionamiento gráficamente en la siguiente imagen:

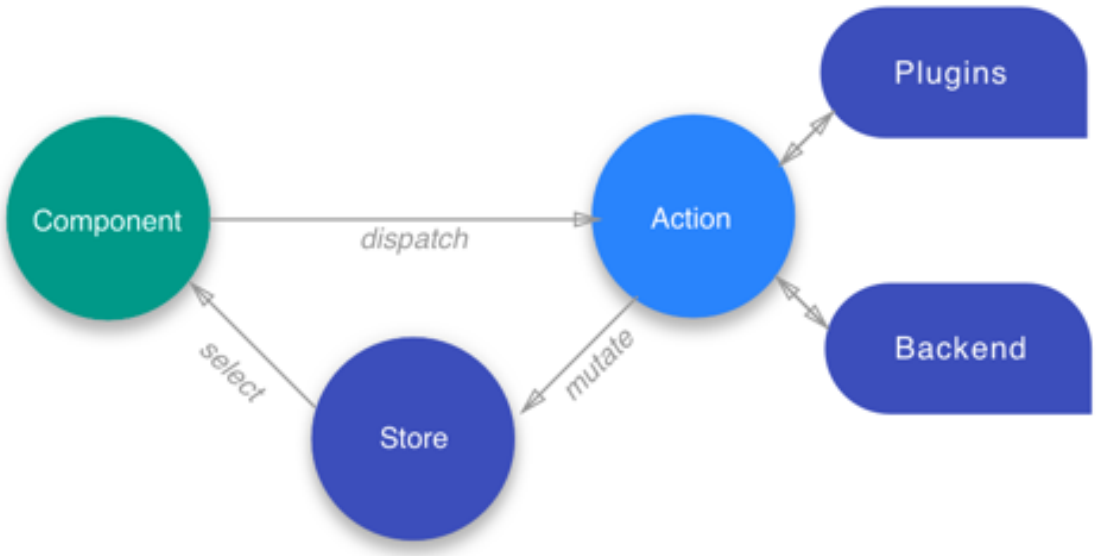


Ilustración 41 Flujo interno de biblioteca NGXS [40]

En angular esto se ve reflejado en la carpeta denominada store que contiene dos scripts:

nombre.action.ts
nombre.state.ts

6.12 Desarrollo – Home y Layout

La página principal en la que se aterriza al iniciar sesión es la *home*, esta está definida como un módulo ubicado dentro de la carpeta *modules* y la ruta con la que se accede es url_raiz/home. También se puede acceder a *home* desde la opción de la barra lateral 'Inicio'.

La estructura de *home* se desarrolla como panel informativo, en el que se muestra la información general, un gráfico circular creado con la biblioteca "ng2-charts" de npm [41], y un grid con tres opciones:

- Añade tipo de deporte
- Añade deporte
- Añade movimiento

Estas opciones actualmente llevan a vacío ya que se espera a desarrollar el resto de módulos para poder redirigirlo a su ruta correspondiente.

El layout principal de este proyecto está creado con la librería "ng-zorro" y es un componente ubicado en la carpeta ui, también se define aquí la vista not-found.

6.13 Desarrollo - Componente Three.js, fbx-loader

Una vez funciona el inicio de sesión y el registro, con aterrizaje en 'home', se decide comenzar por el módulo Three.js para poder visualizar y tener operativo el visor 3D de archivos FBX. Este desarrollo me pareció un tanto complejo ya que no había tratado la librería Three.js desde Angular en lenguaje typescript, esto cambia un poco las cosas, pero gracias a la gran comunidad y el apoyo de los usuarios del fórum [42] de Three.js pude desarrollarla por completo.

El funcionamiento principal de esta librería se basa en la librería WebGL, Web Graphics Library, que es una especificación estándar que define una API implementada en JavaScript para la renderización de gráficos en 3D dentro de cualquier navegador web.

La estructura principal de Three.js se divide en una escena, una cámara y un Renderer.

- Escena, que es como un lienzo donde poder colocar nuestros elementos gráficos.
    - Luces
    - Objetos o grupo de objetos
    - Forma o Mesh, que está compuesto por:
        - Geometría
        - Material que tendrán una textura
- Cámara, enfoca la escena
- Renderer, toma todo lo que esta en la escena bajo el foco de la cámara y lo muestra en un navegador.

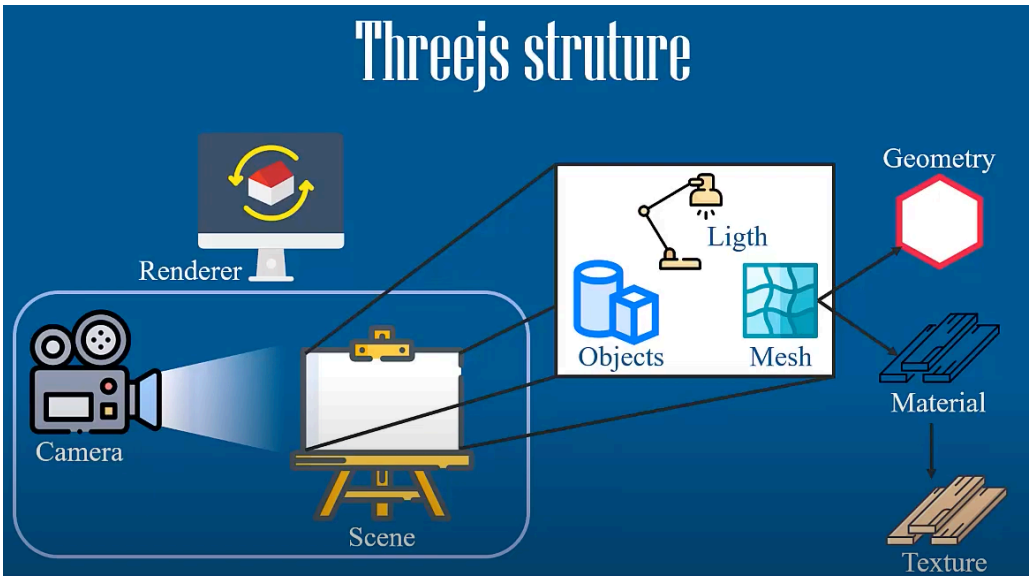


Ilustración 42 Estructura Three.js [43]

Por lo tanto, en el componente al que llamamos "three" se crea un renderer, una escena, y se añaden luces y cámara, y se carga un objeto, en este caso un objeto en formato FBX mediante el 'fbx-loader'. En este componente también se desarrolla el suelo, y un esqueleto auxiliar que marcará el movimiento mediante unos ejes desarrollados en el componente.

A este desarrollo se le añaden unos botones de pausa y play y reset, para controlar la animación.

A la hora de crear el visor se prueba con objetos FBX provenientes de diferentes partes, el primero que se prueba es el de Mixamo [44], que es una plataforma de venta de modelos 3D y animaciones, pero también tiene algunas gratuitas. Estas se ven sin problemas. Posteriormente se prueba con las convertidas con el conversor que hemos desarrollado que exportadas desde el sistema BTS SMART-D en formato mdx a FBX, y este se ve sin problemas.

Los problemas vinieron con la exportación del FBX del sistema Wearnotch, ya que no tenía esqueleto, pero si animación, para ello la solución desarrollada es tener un modelo en la parte pública del proyecto en la carpeta assets, este modelo tiene con un rig estructurado igual que en los archivos descargados de Wearnotch para aplicarlo cuando este se cargue.

A la hora de realizar la implementación del componente 'three' al cargar el movimiento en el constructor desde la 'store' se identifica si este movimiento es de Notch o no, en este caso se activa una variable. Esta variable a la que llamamos 'objectWithoutModel', se emplea dentro del fbx-loader, ya que si está activa lo que se hace es asignar el modelo base a la animación. Este modelo fue realizado con ayuda de un experto en modelado 3D y rigging, el modelo es el mismo que Mixamo pero con el esqueleto y la pose de Wearnotch.

6.14 Desarrollo - Gestor de usuarios

El gestor de usuarios es solo accesible para usuarios con rol ADMIN, y es un módulo al que se le ha llamado 'users', este módulo está compuesto por dos componentes.

6.14.1 Users

Este componente está destinado al control de usuarios, una tabla que lista todos los usuarios que hay registrados en la base de datos, estos se pueden filtrar, ordenar, buscar y eliminar. Desde esta misma vista se pueden crear nuevos usuarios y también se puede refrescar

la tabla de usuarios, esta tabla muestra usuarios paginados de 10 en 10. Y es accesible desde el menú lateral 'Mi Cuenta'. La ruta para acceder a esta vista es 'url_raiz/users'.

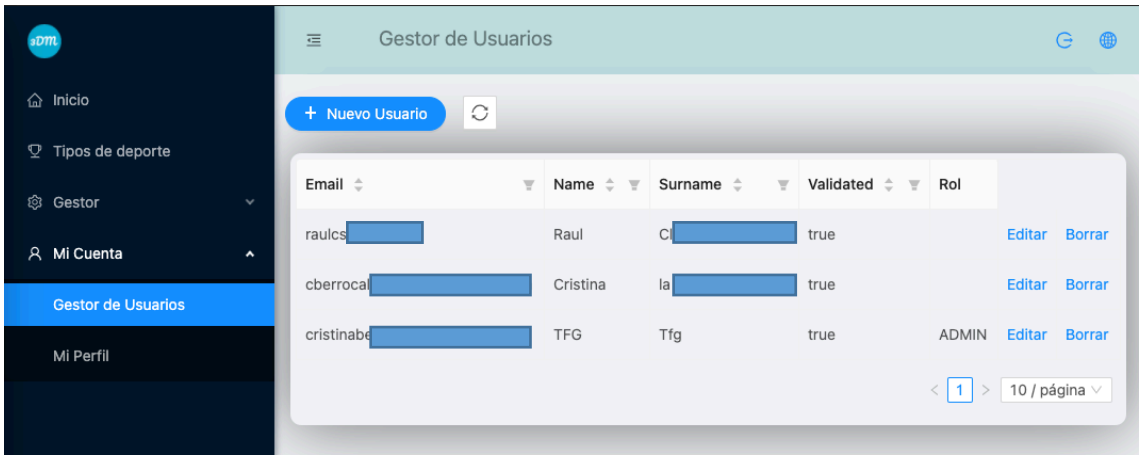


Ilustración 43 Vista de Gestor de Usuarios

6.14.2 User

Una vez seleccionas un usuario para su edición o vista, o quieres crear otro nuevo el componente que se carga es 'user', las rutas para llegar a este componente son:

- url_raiz/users/:id, esta ruta muestra un usuario seleccionado, para su edición o vista.
- url_raiz /users/new, esta ruta te muestra el componente vacío para que sea rellenado para crear un nuevo usuario.

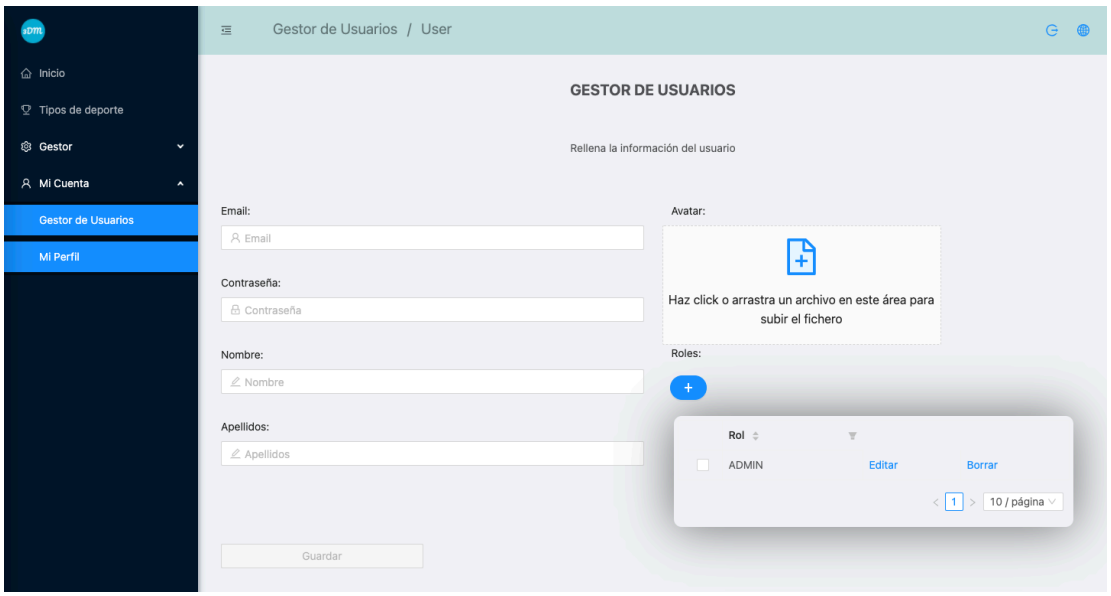


Ilustración 44 Vista usuario de gestor de usuarios

6.15 Desarrollo - Gestor de tipos de deporte, deportes y movimientos

Se puede decir que después de desarrollar el Gestor de usuarios, el resto de los gestores funcionan de una forma similar, la estructura es la misma.

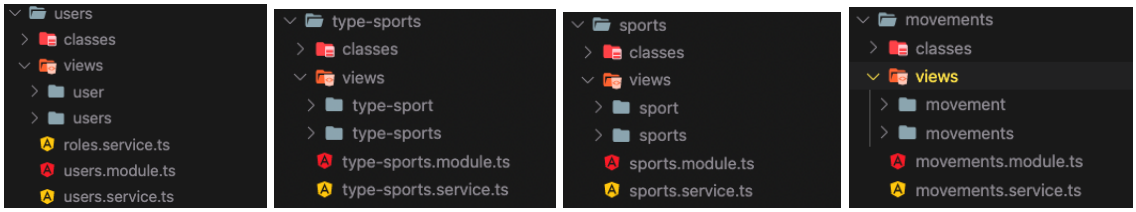


Ilustración 45 Estructura de los gestores

El componente con el nombre en plural (users, sports, …) engloba la tabla en la que se puede buscar, filtrar, ordenar, seleccionar y borrar elementos de la misma. Los componenetes con nombre singular se refieren al formulario como tal, la estrutura que siguen para su acceso es la misma que la de usuario:

Módulo	Rutas	Componente
Type-sport	/type-sports	Lista de todos los type-sports
	/type-sports/:id	Formulario de type-sport relleno de la información
	/type-sports/new	Formulario de type-sport vacío
Sports	/sports	Lista de todos los sports
	/sports/:id	Formulario de sport relleno de la información
	/sports/new	Formulario de sport vacío
Movements	/movements	Lista de todos los movements
	/movements/:id	Formulario de movement relleno de la información
	/movements/new	Formulario de movement vacío

Tabla 3 Rutas de los gestores

Al terminar el desarrollo de los gestores, se puso el link dentro de los botones de *home* para que redirijan al formulario vacío de las rutas acabadas en /new.

6.16 Desarrollo – Visor

El desarrollo del visor se desarrolló como un solo módulo, que engloba diferentes componentes:

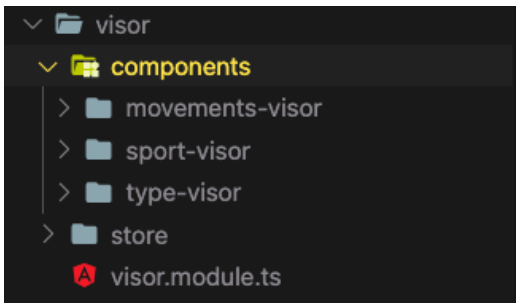


Ilustración 46 Módulo visor

Cada uno es una vista de un visor diferente, pero al compartir el store entre los tres se decidió desarrollar como un solo módulo. Se puede acceder a estos visores de forma escalonada uno después de otro desde la opción de la barra latera 'Tipos de deporte', la ruta es: url_raíz/visor, el primer visor que se carga es el de Tipos de deporte.

6.16.1 Desarrollo – Visor tipo de deportes

El visor de tipos de deporte muestra un grid de los tipos de deporte que hay en la base de datos, con una pequeña descripción e imagen, cada uno de estos tipos de deporte te redirige al visor de deportes del tipo de deporte seleccionado. La url a la que te redirige es la siguiente:

```
url_raiz/visor/sportVisor/:id
```

Este id corresponderá al del tipo de deporte seleccionado.

6.16.2 Desarrollo - Visor de deportes

El visor de deporte muestra un grid de los deportes del tipo de deporte seleccionado que hay en la base de datos, con una pequeña descripción e imagen, cada uno de estos deportes te redirige al visor de deportes del deporte seleccionado. La url a la que te redirige es la siguiente:

```
url_raiz/visor/movementsVisor/:id
```

Este id corresponderá al deporte seleccionado.

6.16.3 Desarrollo - Visor de movimientos

El visor de movimientos muestra un grid de los movimientos del deporte seleccionado que hay en la base de datos, con una pequeña descripción e imagen, cada uno de estos movimientos te redirige al visor 3D del movimiento seleccionado. La url a la que te redirige es la siguiente:

```
url_raiz/visor/movement/:id
```

Este id corresponderá al movimiento seleccionado.

6.17 Puesta en producción

A la hora de realizar el despliegue de la aplicación web supuso solucionar una serie de problemas:

- Instalar Apache
- Lanzar front-end
- Instalar Mysql
- Instalar pm2
- Lanzar back-end

6.17.1 Instalar Apache

Se establece la conexión a través de la terminal, mediante conexión SSH e instalamos el paquete de Apache2:

```
sudo apt update
sudo apt upgrade
sudo apt install apache2
```

Al instalar este paquete el servidor Apache ya funciona, podemos comprobar su funcionamiento mediante el comando *sudo service apache2 status* en la consola. También se puede acceder a la IP de nuestro servidor en el navegador para poder ver la página por defecto de Apache.

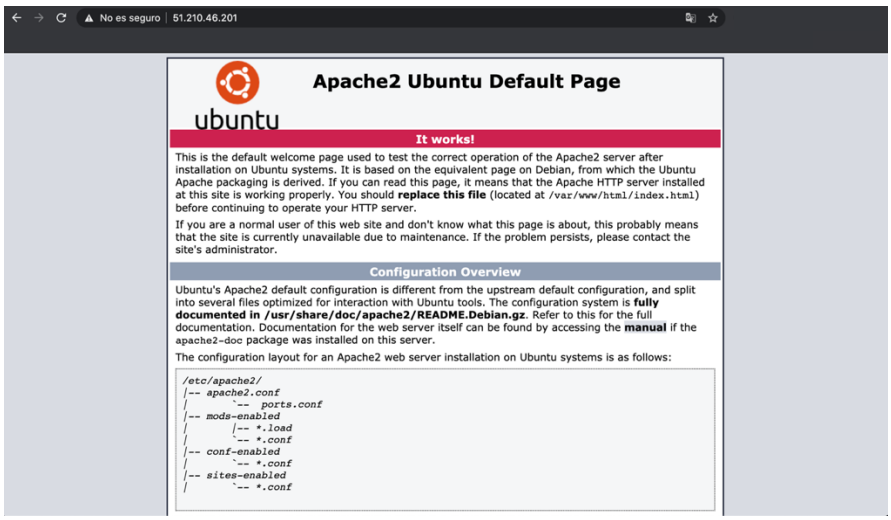


Ilustración 47 Apache instalado en el VPS

6.17.2 Lanzar front-end

Para desplegar el frontal lo hago a través de Filezilla [45] que es una aplicación FTP libre de código abierto que consta de un cliente y un servidor, en este caso la conexión que utilizo es SFTP.

Cada vez que me conecto a FileZilla y quiero renovar el frontal traspaso el contenido de la carpeta dist generada del 'npm run build --prod' del proyecto a la carpeta var/www/html y sustituyo los archivos después reinicio el apache2, con el comando 'sudo service apache2 restart'.

6.17.3 Instalar Mysql

Instalo MySQL a través de la terminal, creo el usuario, le doy privilegios y creo el esquema de la base de datos, configuro el proyecto con la conexión y los nuevos datos de usuario.

6.17.4 Instalar pm2

Para poder instalar bien pm2 se siguen los siguientes pasos:

1. Install Yarn
2. Install PM2
3. Uso Git para hacer un fetch del proyecto NestJs desde Github en home/app-tfg
4. Checkout a la rama develop
5. Configuro el environrment delk proyecto NestJs con la nueva conexión al VPS
6. Compilo el Proyecto por linea de comantos, 'npm run build'
7. Para ejecutarlo, hay que ejecutar con pm2 la carpeta main/dist.js.

Al desplegar el proyecto NestJs por primera vez se generarán en la base de datos todas las tablas necesarias.

6.17.5 Lanzar back-end

Para poder desplegar en otras ocasiones el back-end los pasos a seguir son los siguientes:

1. Moverte a la carpeta contenedora del proyecto NestJs.
2. Descargar el proyecto de Git, o transferir los archivos mediante Filezilla.
3. Compilar mediante línea de comandos.
4. Parar Pm2.
5. Volver a reiniciar pm2.

Un ejemplo de despliegue por línea de comandos se puede ver a continuación, previamente a esto, había traspasado mediante Filezilla la última versión del proyecto.

```
Documentos — ubuntu@vps-dafb03bd: /home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy — ssh ubuntu@51.210.46.201 — 148×43
~/Documentos — ubuntu@vps-dafb03bd: /home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy — ssh ubuntu@51.210.46.201
  Super simple, hardened and opinionated Kubernetes for production.

    https://microk8s.io/high-availability

67 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Wed Nov 18 15:37:03 2020 from 188.78.183.41
ubuntu@vps-dafb03bd:~$ cd ..
ubuntu@vps-dafb03bd:/home$ cd app-tfg/
ubuntu@vps-dafb03bd:/home/app-tfg$ cd netsjs-from-3D-movement-to-biomechanical-pedagogy/
ubuntu@vps-dafb03bd:/home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy$ sudo npm run build

> nestjs-3dmovement-to-biomechanical@0.0.1 prebuild /home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy
> rimraf dist


> nestjs-3dmovement-to-biomechanical@0.0.1 build /home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy
> nest build

ubuntu@vps-dafb03bd:/home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy$ sudo pm2 stop "back-nestjs"
[PM2] Applying action stopProcessId on app [back-nestjs](ids: 0)
[PM2] [back-nestjs](0) ✓
| id | name | namespace | version | mode | pid | uptime | ↺ | status | cpu | mem | user | watching |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | back-nestjs | default | 0.0.1 | fork | 0 | 0 | 1 | stopped | 0% | 0b | root | disabled |
ubuntu@vps-dafb03bd:/home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy$ sudo pm2 restart "back-nestjs"
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [back-nestjs](ids: 0)
[PM2] [back-nestjs](0) ✓
| id | name | namespace | version | mode | pid | uptime | ↺ | status | cpu | mem | user | watching |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | back-nestjs | default | 0.0.1 | fork | 1578236 | 0s | 1 | online | 0% | 18.1mb | root | disabled |
ubuntu@vps-dafb03bd:/home/app-tfg/netsjs-from-3D-movement-to-biomechanical-pedagogy$ sudo pm2 log "back-nestjs"
[TAILING] Tailing last 15 lines for [back-nestjs] process (change the value with --lines option)
/root/.pm2/logs/back-nestjs-error.log last 15 lines:
```

Ilustración 48 Ejemplo de despliegue en producción de back-end

6.18 Problemas encontrados

Una vez el proyecto estaba acabado y subido a producción antes de poder terminar la memoria y defenderlo, me encontré con una situación inesperada. El 10 de marzo de 2021 salió en los medios de comunicación el incendio de uno de los edificios de la empresa OVH, en concreto el edificio SBG2, en Estrasburgo, Francia. Tras ver las noticias procedo a ver el acceso de mi servidor, accedo a la IP de mi VPS y no hay respuesta, el servidor ha caído. En este momento volví a la calma tras un breve período de crisis y me decidí a esperar. El 12 de marzo recibí un correo de OVH que decía:

"Si su VPS está ubicado en el centro de datos SBG3 (correspondiente a la ubicación del VPS SBG6), podrá volver a ponerlo en servicio a partir del 22 de marzo."

El 22 de marzo, el VPS volvió a estar operativo, pero al entrar en la IP a través del navegador este no funcionaba correctamente, la solución que apliqué fue recompilar el proyecto completo y volverlo a desplegar, en este momento todo volvió a la normalidad. Creo que tuve mucha suerte y con paciencia todo se arregla. Otras compañías y servidores alojados en otros edificios no sufrieron la misma suerte ya que sus servicios no han podido ser reactivados y han sido perdidos.

Me enfrenté a este problema sin temor ya que poseía una copia de seguridad en local y en GitHub.

7 Estudio económico

Para realizar el estudio económico de este proyecto se calcula el coste de los recursos involucrados en el mismo. Por un lado, se tendrán en cuenta los recursos humanos, los recursos materiales, las licencias y los costes indirectos, basando los datos en el precio hora propuesto por el trabajador como cuota de *Freelancer*. Se detalla el presupuesto a continuación.

7.1 Presupuesto

Al realizar el presupuesto se ha valorado el trabajo realizado por la autora del PFG a lo largo de todas las fases del desarrollo. Es cierto que el rol principal es el de programadora, pero a lo largo del desarrollo se han desempeñado tareas pertenecientes a los roles de diseñador y analista, por lo que el coste de recursos humanos se ha estimado según el rol desempeñado en cada hora.

7.1.1 Costes de recursos humanos

Se ha planteado este proyecto como un proyecto *Freelancer* por lo que las cuotas y tasas son fijadas por el trabajador, siguiendo un precio destinado a cada tipo de trabajo. La distribución de horas en este proyecto se ve distribuida de la siguiente manera tal y como podemos ver en la Ilustración 49.

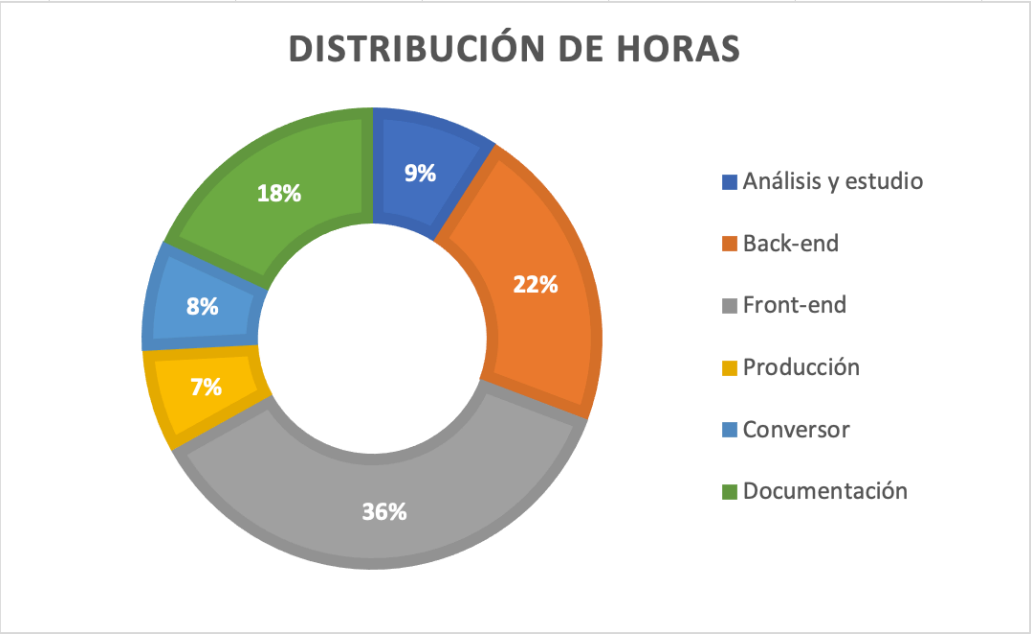


Ilustración 49 Distribución de horas en el proyecto

El precio de cada hora y las horas totales quedan por lo tanto de la siguiente manera:

Tareas		Horas de trabajo (h)	Precio €/h	Total
Análisis y estudio		35	20	700 €
Back-end	Diseño	14	30	420 €
	Implementación	70	20	1.750 €
Front-end	Diseño	35	30	1.050 €
	Implementación	105	20	2.625 €
Producción		28	25	840 €
Conversor		30	35	1.050 €
Documentación		70	15	1.050 €
Total		387 h	-	9.485 €

Tabla 4 Costes de recursos humanos

7.1.2 Costes de recursos materiales

En segundo lugar, se calcula los costes de los materiales amortizados. En este caso, hay una serie de materiales que han servido para el desarrollo del proyecto, y los detallamos a continuación:

	Precio	Vida útil	Duración mes	Coste por mes	Total
Macbook Pro 13"	1500€	5 años [46]	2,6	25€	65€
Periféricos	150€	6 años [47]	2,6	2,08€	5,41€
Pantalla 24" x 2	140€	3,4 años [48]	2,6	3,43€	8,92€

Tabla 5 Costes de recursos materiales

El total de costes por recursos materiales asciende a 79,33€.

7.1.3 Costes de licencias de software

En tercer lugar, se tiene en cuenta el coste de las licencias de software empleadas. En este caso no se va a añadir ningún coste debido a los sistemas operativos ya que vienen incluidos dentro del coste de los materiales. Los costes debidos a las licencias de software son los siguientes:

	Precio por mes	Tiempo de empleo (meses)	Total
VPS	10 € / mes	2,6	26 €
Office	8.80 € / mes	2,6	22.88 €
		Coste total licencias de software	48.88 €

Tabla 6 Costes de licencias de software

7.1.4 Costes totales

En último lugar, se hace un recuento de todos los costes, añadiendo un 20% por costes indirectos, que bien pueden ser costes de electricidad, internet, etc.

El presupuesto inicial se detalla a continuación.

	TOTAL
Recursos humanos	9.485 €
Recursos materiales	79,33 €
Licencias de software	48.88 €
Total costes directos	9.613.21 €
Total costes indirectos (20%)	1.922.65 €
Presupuesto total	11.535,85 €

Tabla 7 Presupuesto

Como se puede ver en la tabla anterior, los costes totales de este proyecto ascienden a 11.535,85 €.

8 Resultados y discusión

El resultado final del proyecto es una aplicación web cuya finalidad es ser una plataforma de uso educativo y de almacenamiento, para proporcionar acceso a diferentes usuarios de forma remota.

Como principales resultados de este proyecto de fin de grado cabe decir que se han cumplido con los objetivos definidos del mismo, y ha sido un resultado satisfactorio cumpliendo con las necesidades básicas proporcionada como versión inicial, y puesta en marcha de la idea.

A la hora de hablar de los resultados se va a dividir en tres secciones bien diferenciadas que nos van a hacer comprender el resultado del proyecto.

8.1 Cumplimiento de los objetivos

Respecto al grado de cumplimiento de los objetivos de este proyecto, se puede decir que han sido cumplidos en su totalidad.

Se ha creado un sistema de conversión de ficheros de mo-cap (motion capture) del formato mdx a otros que pueden ser visualizados fácilmente en web como es el FBX.

Se ha creado una base de datos para el almacenamiento de estos ficheros y de toda la información relativa y necesaria al proyecto web, y con este se ha automatizado el proceso para añadir estos ficheros conforme se van creando, mediante el gesto de movimientos.

Se ha diseñado y creado una web con una parte de administrador para poder gestionar fácilmente la base de datos mediante los gestores, en esta web hay una implementación del visualizador 3D con el que se puede interactuar para poder mover los objetos, pausarlo y reiniciarlo, para poder hacer uso pedagógico o de investigación con él.

8.2 Limitaciones

El resultado de la plataforma demuestra que es un proyecto viable y que puede ser escalable con facilidad. Con una buena planificación de mejoras y tiempo puede llegar a ser un proyecto muy útil y viable también en la vida real.

9 Conclusiones

Concluyo que la plataforma de almacenamiento de movimientos deportivos capturados mediante *motion-capture* y su clasificación son viables. Este proyecto es una versión beta, la creación de una idea inicial a desarrollar, la intención inicial ha sido poder cumplir con los objetivos señalados y demostrar que la idea era viable. De este modo queda mucho desarrollo por realizar para que esta plataforma web sea de calidad y esté optimizada.

9.1 Conocimientos adquiridos

Como conocimientos adquiridos he podido diseñar, planificar y crear un proyecto completo desde cero, partiendo de una idea propia y de un cliente, creo que este trabajo me ha dotado de perspectiva y he podido experimentar con nuevas tecnologías con las que tenía muchas ganas de trabajar, he podido aprender de mis errores y aciertos, creo que la parte más compleja ha podido ser el tiempo, ya que realizar este proyecto mientras se desempeña una actividad laboral a jornada completa no es tan sencillo, y pasa bastante tiempo entre fases de desarrollo a veces ya no te acuerdas por dónde vas, así que tomé la costumbre de comentar el código y dejarme escrito el status del proyecto, estas buenas prácticas me han hecho mejorar como desarrolladora.

9.2 Propuesta de mejora

Algunas de las mejoras pendientes son las siguientes:

- Hacer un diseño *responsive*, ya que con el se busca la correcta visualización de una misma página en distintos dispositivos. Y compilarlo para Android e iOS añadiendo al proyecto Capacitor, que es un framework que permite "transformar" una aplicación web en una aplicación móvil o de escritorio mediante el uso de WebView.
- Mejorar el control de usuarios dotándolos de una mayor seguridad y una mejor estructura para que pueda ser escalable y útil para cualquier cliente no solo la Universidad San Jorge. De este modo el proyecto estaría en un contenedor y cada entidad tendría su propia base de datos con la misma estructura, el front-end sería compartido por todos los clientes. De esta manera cualquier usuario tendría acceso remoto a su plataforma sin peligro de cruce de datos. En este punto cabría valorar la idea de que quizá el cruce de datos sea una ventaja a la hora de generalizar la aplicación, meter todos los datos en una base de datos, ya que permitiría tener una base de datos de movimientos más amplia y, por tanto, la colaboración entre distintas entidades.
- Huesos del esqueleto seleccionables que marquen líneas con su paso y que estas sean exportables en gráficos 2D seleccionando los ejes.

- Realizaría Test unitarios, para comprobar la robustez de la aplicación y pruebas de estrés en el servidor, para comprobar la capacidad de este y su latencia.
- Implementación de cambio de modelo 3D en el visualizador, para poder poner el modelo de una persona con sobrepeso o seleccionar género.
- Integrar el conversor en la web, para que al subir los archivos del formato que sea se conviertan en FBX directamente.
- El conversor, realizar un conversor completo y robusto de algunos formatos habituales a FBX podría ser un proyecto de fin de grado completo, ya que es una tarea compleja por la falta de documentación y que podría ser útil para una comunidad muy amplia.
- Terminar las traducciones de la aplicación con la biblioteca Transloco [49].
- Hacer una base de datos con traducciones

10 Bibliografía

[1] «Moodle,» [En línea]. Available: https://moodle.org/.

[2] «Edmodo,» [En línea]. Available: https://new.edmodo.com/.

[3] Google, «Administra la enseñanza y el aprendizaje con Classroom,» Google, [En línea]. Available: https://edu.google.com/intl/es-419/products/classroom/?modal_active=none&gclid=Cj0KCQiA9P__BRC0ARIsAEZ6iri8eCeQhjW681wPTmq7rWO-PzsaDVRhFcOqFV1iIOp_jEXVOgEMLOUaAq-jEALw_wcB.

[4] Google, «Google Forms,» [En línea]. Available: https://docs.google.com/forms.

[5] Angular, «Angular,» [En línea]. Available: https://angular.io/.

[6] ThreeJs, «ThreeJs,» [En línea]. Available: https://threejs.org/.

[7] model-viewer, «ModelViwer,» [En línea]. Available: https://modelviewer.dev/.

[8] btsbioengineering, «btsbioengineering,» [En línea]. Available: https://www.btsbioengineering.com/es/products/bts-sportlab-motion-capture-sport/.

[9] Noraxon, «Noraxon,» [En línea]. Available: https://www.noraxon.com/our-products/myomotion/.

[10] Wearnotch, «Wearnotch,» [En línea]. Available: https://wearnotch.com/developers/docs/sdk/manual/.

[11] B. Bioengineering, SMART-D User Manual, 2008.

[12] BTS Bioengineering, «BTS SMART-DX High-precision optoelectronic system for the biomechanical motion analysis,» [En línea]. Available: http://www.arrayamed.com/fullaccess/product62file1.pdf. [Último acceso: Enero 2021].

[13] C3D, «C3D,» [En línea]. Available: www.c3d.org.

[14] Noraxon, «Noraxon.com,» Noraxon.com, 2017. [En línea]. Available: https://www.noraxon.com/wp-content/uploads/2017/10/NRXN_P-4638_Rev_A_3-17_F5-2.pdf.

[15] Noraxon, «myoMOTION™ Software Module,» Noraxon, [En línea]. Available: https://www.noraxon.com/our-products/myomotion/.

[16] Notch, «Importing Custom Skeleton (BVH),» Notch, [En línea]. Available: https://docs.wearnotch.com/docs/pioneerapp_import_bvh/.

[17] NestJs, «NestJs,» [En línea]. Available: https://nestjs.com/.

[18] Unity, «Unity,» [En línea]. Available: https://unity.com/es.

[19] OVHCloud, «VPS de OVH,» [En línea]. Available: https://www.ovhcloud.com/es-es/vps/.

[20] OVHCloud, «Hosting OVH,» [En línea]. Available: https://www.ovh.es/hosting/hosting-personal.xml?_gl=1*mhsbym*_gcl_aw*R0NMLjE2MDMwMTk4MzAuQ2p3S0NBand6Nl84QlJCa0Vpd0EzcDAyVmJES0NGSTdQZmtkemlxcy1ZU1p2ZF9iY09LVV9jWTBfV2lYTVVCdFZiZ2F2VmpuRHdYN0lCb0NqdjhRQXZEX0J3RQ...

[21] Apache, «Apache,» [En línea]. Available: https://httpd.apache.org/.

[22] Pm2, «Pm2,» [En línea]. Available: https://pm2.keymetrics.io/.

[23] D. Angular, «Documentación Angular,» [En línea]. Available: https://angular.io/.

[24] Nest, «Documentación NestJs,» [En línea]. Available: https://docs.nestjs.com/.

[25] Nest, «Repositorio Nest,» [En línea]. Available: https://github.com/nestjs/nest/tree/master/sample.

[26] Chrome, «V8,» [En línea]. Available: https://v8.dev/.

[27] SmartBear, «SmartBear Software,» [En línea]. Available: https://smartbear.com/?utm_source=aw&utm_medium=ppcg&utm_term=smartbear%20software&utm_content=447022573884&utm_campaign=9507841160&gclid=CjwKCAiAi_D_BRApEiwASslbJ1XGIhR3PCgWaaYgWQVrQ1bgZN9au5_xtxtprEiLjkVNj9v9OI85hRoCLogQAvD_BwE&gclsrc=aw.ds.

[28] lcruz, «¿Que es RESTful Web Service? y ¿qué es REST?,» 8 Febrero 2021. [En línea]. Available: https://codigonaranja.com/restful-web-service.

[29] OVH, «OVH cloud,» [En línea]. Available: https://www.ovh.es/.

[30] Trello, «Trello,» [En línea]. Available: https://trello.com/.

[31] Git. [En línea]. Available: https://git-scm.com/.

[32] GitHub, «GitHub,» [En línea]. Available: https://github.com/.

[33] ExpressJS, «ExpressJS,» [En línea]. Available: https://expressjs.com/es/.

[34] Microsoft, «Microsoft Excel,» [En línea]. Available: https://www.microsoft.com/es-es/microsoft-365/excel.

[35] E. Serra, «3dart,» 31 Diciembre 2016. [En línea]. Available: https://www.3dart.it/en/download-free-rigged-deadpool-3d-model/.

[36] Blender, «Blender,» [En línea]. Available: https://www.blender.org/about/.

[37] Nestjs, «OpenAPI Types and parameters,» [En línea]. Available: https://docs.nestjs.com/openapi/types-and-parameters#types-and-parameters.

[38] MySQL, «MySQL Workbench,» [En línea]. Available: https://www.mysql.com/products/workbench/.

[39] L. M. Lopez Magaña, «Qué es Json Web Token y cómo funciona,» OpenWebinars, 17 Enero 2020. [En línea]. Available: https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/.

[40] NGXS, «NGXS,» [En línea]. Available: https://www.ngxs.io/.

[41] ng2-charts, «Npmjs,» [En línea]. Available: https://www.npmjs.com/package/ng2-charts.

[42] three.js, «discourse.threejs.org,» [En línea]. Available: https://discourse.threejs.org/.

[43] MonkeyWit, «Youtube.com,» [En línea]. Available: https://www.youtube.com/watch?time_continue=168&v=VxAH-c1ueR4.

[44] Mixamo, «Mixamo,» [En línea]. Available: https://www.mixamo.com/.

[45] Filezilla, «Filezilla,» [En línea]. Available: https://filezilla-project.org/.

[46] R. TICPymes, «Ticpymes,» 04 Abril 2019. [En línea]. Available: https://www.ticpymes.es/tecnologia/noticias/1111246049504/vida-util-de-tecnologia-apple-puede-llegar-12-anos.1.html#:~:text=La%20vida%20%C3%BAtil%20de%20la.

[47] R. Castro, «Wikiversus,» 19 Abril 2020. [En línea]. Available: https://www.wikiversus.com/informatica/cuanto-tiempo-equivalen-millones-de-pulsaciones/.

[48] J. D. d. Usera, «Hardzone,» 13 Enero 2019. [En línea]. Available: https://hardzone.es/2019/01/13/vida-util-monitor-antes-rompa/.

[49] Transloco. [En línea]. Available: https://ngneat.github.io/transloco/.

[50] Node.js, «Node.js,» [En línea]. Available: https://nodejs.org/es/.

[51] «Preguntas frecuentes,» USJ, [En línea]. Available: https://www.usj.es/alumnos/practicas/preguntas-frecuentes.

[52] NestJs, «Passport Authentication,» [En línea]. Available: https://docs.nestjs.com/security/authentication.

Anexo I – Propuesta del proyecto

Nombre alumno:	Cristina Berrocal
Titulación:	Graduado en Diseño y Desarrollo de Videojuegos
Curso académico:	2020-2021

1. TÍTULO DEL PROYECTO

Del movimiento 3D a la pedagogía biomecánica.

2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TEMA A TRATAR

Desarrollar una aplicación informática que permita la automatización del proceso de conversión y adaptación de ficheros de captura de movimiento 3D de movimientos deportivos a un formato que permita ser archivado en una base de datos y, la aplicación web que permita la visualización de estos ficheros junto con las herramientas necesarias para su uso tanto de forma pedagógica como para investigación.

3. OBJETIVOS DEL PROYECTO

Los objetivos del proyecto son:

- Creación del sistema de conversión de ficheros de mo-cap (motion capture) a otros que puedan ser visualizados fácilmente en web.
- Creación de una base de datos para su almacenamiento y automatización del proceso para añadir estos ficheros conforme se van creando.
- Creación de una web con una parte de administrador para poder gestionar fácilmente la base de datos.
- Creación de una herramienta web que permita la visualización e interacción con las grabaciones de movimiento 3D para uso pedagógico o de investigación.

4. METODOLOGÍA

La metodología se establecerá en las primeras fases del proyecto.

5. PLANIFICACIÓN DE TAREAS

Las tareas quedan predefinidas de manera global en los objetivos. Serán fijadas de forma concreta durante el desarrollo del proyecto.

6. OBSERVACIONES ADICIONALES

Profesor: Eduardo Jiménez, y Vanessa Bataller (Facultad de Salud).

Anexo II – Reuniones

Reunión:	**1**		
Fecha:	5/06/19		
Hora comienzo:	12:00	**Hora finalización:**	13:00
Lugar:	Universidad San Jorge, Planta Baja Edificio 3 Salud		
Elabora acta:	Cristina Berrocal		
Convocados:	Vanessa Bataller		
Orden del día	Definir el PFG		

No	Asunto
1	Motion capture, se plantea hacer mediante *Motion Capture* la captura de los esqueletos de las diferentes técnicas básicas de las diferentes disciplinas para hacer un repositorio en los servidores de la universidad. Hay que buscar a finalidad si será una app o una web.
2	Próxima reunión a demanda

Reunión:	**2**		
Fecha:	5/06/19		
Hora comienzo:	13:30	**Hora finalización:**	14:30
Lugar:	Universidad San Jorge, Facultad Rectorado		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Planificar el inicio del PFG		

No	Asunto
1	Se definen objetivos la metodología queda ha decisión de la desarrolladora, hay que ver si va a ser web o app después de la toma de requisitos, hay que identificar el stack tecnológico porque variará el entorno y los lenguajes de programación. Hay que hacer un análisis previo de requisitos y un diseño del frontal.
2	Próxima reunión a demanda

Reunión:	**3**		
Fecha:	19/06/2019		
Hora comienzo:	13:00	**Hora finalización:**	14:00
Lugar:	Universidad San Jorge, laboratorio de Salud.		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez y Vanessa Bataller		
Orden del día	Conocerse y hablar sobre el proyecto y sus objetivos		

No	Asunto
1	Esta reunión sirve para comprender los requisitos del cliente, en este caso Vanessa Bataller, y hablar sobre el proyecto y los sistemas de captura.
2	Próxima reunión a demanda

Reunión:	**4**		
Fecha:	24-27/04/2020		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	Correo Outlok		
Elabora acta:	Cristina Berrocal		
Convocados:	Vanessa Bataller		
Orden del día	Diseño de la web y exportaciones		

No	Asunto
1	Se intercambian una serie de correos para la confirmación del Diseño de la web y una solicitud de exportaciones para su correspondiente análisis y diseño del conversor.
2	Próxima reunión a demanda

Reunión:	**5**		
Fecha:	7/07/2020		
Hora comienzo:	19:30	**Hora finalización:**	20:30
Lugar:	On-line, Skype		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Conversor y formatos		

No	Asunto

1	Esta reunión es destinada al diseño y desarrollo del conversor.
2	Próxima reunión a demanda

Reunión:	**6**		
Fecha:	8-10/07/2020		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	Correo Outlook		
Elabora acta:	Cristina Berrocal		
Convocados:	Vanessa Bataller		
Orden del día	Status y solicitud de esqueletos para el conversor		
No	**Asunto**		
1	Se intercambian una serie de correos para poder esclarecer la jerarquía de los huesos del esqueleto, así como sus nombres por parte del sistema BTS.		
2	Próxima reunión a demanda		

Reunión:	**7**		
Fecha:	14/07/2020		
Hora comienzo:	18:00	**Hora finalización:**	19:00
Lugar:	On-line, Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Conversor acabado		
No	**Asunto**		
1	Se revisa la versión estable del conversor, funciona con las capturas de tren inferior.		
2	Próxima reunión a demanda		

Reunión:	**8**		
Fecha:	27/07/2020		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	On-line, Mensajes Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		

Orden del día	Status del proyecto
No	**Asunto**
1	Se intercambian un par de mensajes mediante Discord para poner al día sobre el avance del estado del proyecto. Se ha conseguido cargar los FBX exportados en un proyecto de pruebas del frontal, obteniendo información y ayuda de un foro de la comunidad de angular y three.js. En cuanto a la web se está finalizando la conexión a la bbdd y se piensa comenzar con la configuración del VPS.
2	Próxima reunión a demanda

Reunión:	**9**		
Fecha:	27/08/2020		
Hora comienzo:	17:00	**Hora finalización:**	18:00
Lugar:	On-line, Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Status web		
No	**Asunto**		
1	Se enseñan los avances de la web y la base de datos. Se queda que en cuanto este desplegado en Producción se actualice el status de la web mediante mensaje o conferencia.		
2	Próxima reunión a demanda		

Reunión:	**10**		
Fecha:	10/12/2020		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	On-line, Mensajes por Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Status web en producción		
No	**Asunto**		
1	Se intercambian mensajes mediante la plataforma Discord, de cómo acceder a la web, la IP y las contraseñas para iniciar sesión. Eduardo proporciona feedback sobre el resultado.		

	Se establece que en cuanto se hagan estos arreglos se vuelva a quedar,

Reunión:	**11**		
Fecha:	16/12/2020		
Hora comienzo:	12:00	**Hora finalización:**	13:00
Lugar:	On-line, Teams		
Elabora acta:	Cristina Berrocal		
Convocados:	Vanessa Bataller		
Orden del día	Status web en prod y feedback, conversor de tren inferior		
No	**Asunto**		
1	Se comparte y se hace una demostración con Vanessa Bataller sobre la web en producción, para que de su opinión sobre el producto y cómo está quedando. Vanessa comenta varios cambios para el visor de movimientos, el resto de la web le parece bien. Se solicitan nuevas exportaciones para probar el conversor de tren superior. Debido a la próxima maternidad de Vanessa Bataller, se establece un nuevo contacto con el laboratorio, será Jacobo de Renteria de la Peña. Se habla sobre el futuro del proyecto una vez acabado y su aplicación a la vida Universitaria.		
2	Próxima reunión a demanda		

Reunión:	**12**		
Fecha:	18/01/2021		
Hora comienzo:	18:00	**Hora finalización:**	20:00
Lugar:	On-line, Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Status web con producción y problemas con Wearnotch		
No	**Asunto**		
1	En esta reunión se presentan los nuevos cambios del frontal, y los nuevos problemas con las exportaciones de Wearnotch, estas exportaciones contienen solo animación y no modelo, se encuentra una solución a este problema y se estable una guía para poder solventarlo.		

	Se detallan los puntos que quedan por solventar para dar por finalizada la parte de desarrollo del proyecto que nos ocupa.
2	Próxima reunión a demanda

Reunión:	**13**		
Fecha:	20/01/2021		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	On-line, Mensajes por Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Status y producción acabada.		
No	**Asunto**		
1	Se intercambian varios mensajes en la plataforma Discord sobre el estado del proyecto, ya se han solventado los problemas con las exportaciones de Wearnotch, por lo que se da por finalizado el desarrollo práctico del proyecto. Se detalla que solo queda terminar la memoria del PFG, y se hablan sobre algunos comentarios en la versión dos ya entregada.		
2	Próxima reunión a demanda		

Reunión:	**14**		
Fecha:	04/06/2021		
Hora comienzo:	-	**Hora finalización:**	-
Lugar:	On-line, Mensajes por Discord		
Elabora acta:	Cristina Berrocal		
Convocados:	Eduardo Jiménez		
Orden del día	Memoria finalizada.		
No	**Asunto**		
1	Se intercambian varios mensajes en la plataforma Discord sobre el estado de la memoria, se resuelven dudas sobre algunos comentarios en la última versión y se da por finalizada.		
2			

Anexo III - Documentación adjunta

La documentación adjunta (disponible en la PDU) consta del código front-end, back-end y conversor de este Proyecto de fin de grado.

Anexo IV – Encuesta

TFG - Del movimiento 3D a la Biomecánica

Hola soy Cristina Berrocal y esta es una encuesta breve sobre tu opinión personal al respecto de los repositorios/bibliotecas 3D web y su uso como herramientas digitales para el apoyo en el proceso de enseñanza-aprendizaje.

Agradezco de antemano tu tiempo para colaborar en mi Proyecto de Fin de Grado de Diseño y desarrollo de Videojuegos.

Los datos recogidos en este formulario serán utilizados de manera estadística protegiendo la identidad de aquellos que lo completen. Se realiza para fines meramente académicos.

Nota Informativa:

¿En qué consiste este repositorio/biblioteca 3D web de movimiento físicos?

Un repositorio o biblioteca 3D web, consiste en un almacenamiento online de movimientos 3D capturados con cámaras, y ordenado por categorías.

Ejemplo de movimiento 3D, minuto 1:40

Nombre y Apellidos o Anónimo *

Texto de respuesta corta

¿Qué has estudiado? *

1. Enfermería
2. Fisioterapia
3. Medicina
4. Ciencias de la actividad física y del deporte
5. Otro

En caso de otro cuál?

Texto de respuesta corta

¿Te dedicas a la educación? *

- Si
- No

En qué rama impartes materia? *

1. Educación primaria
2. Educación secundaria
3. Formación profesional
4. Educación universitaria
5. Tecnico deportivo, monitor o entrenador.
6. No soy educador

Tienes algún conocimiento sobre biomecánica? *

- Sí
- No

¿Estudiaste biomecánica como asignatura? *

- Si
- No

En caso de haber estudiado biomecánica, qué grado de dificultad tuviste a la hora del aprendizaje sobre la asignatura de biomecánica? Siendo muy fácil 1 muy difícil 10

1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	○	○

Qué grado de utilidad otorgas a las herramienta de aprendizaje digital para el proceso de enseñanza-aprendizaje? Siendo 1 completamente innecesarias y 10 completamente necesarias *

1 2 3 4 5 6 7 8 9 10

Crees que un repositorio o biblioteca de patrón de movimientos básicos podría haber ayudado a tu proceso de enseñanza-aprendizaje? *

- Sí
- No

Crees que puede ser útil para el proceso de enseñanza-aprendizaje en el ámbito deportivo y de salud tener acceso a un repositorio 3D de movimientos? *

- Sí
- No

Utilizarías esta idea para el proceso de enseñanza-aprendizaje en el ámbito del deporte y la salud? *

- Sí
- No
- Tal vez

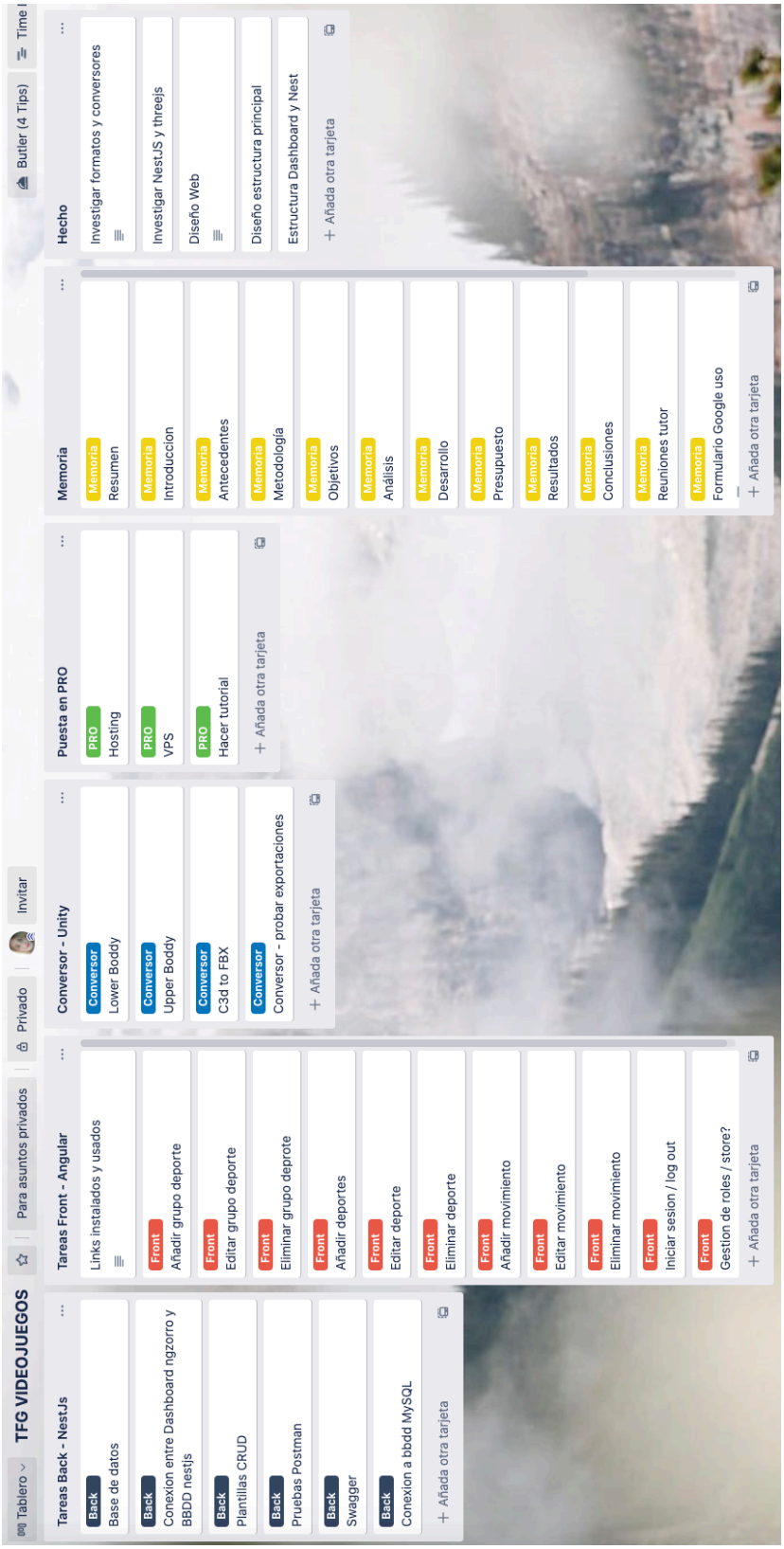
¿Cómo evalúas esta idea? Siendo el 1 muy mala y 10 muy buena. *

1 2 3 4 5 6 7 8 9 10

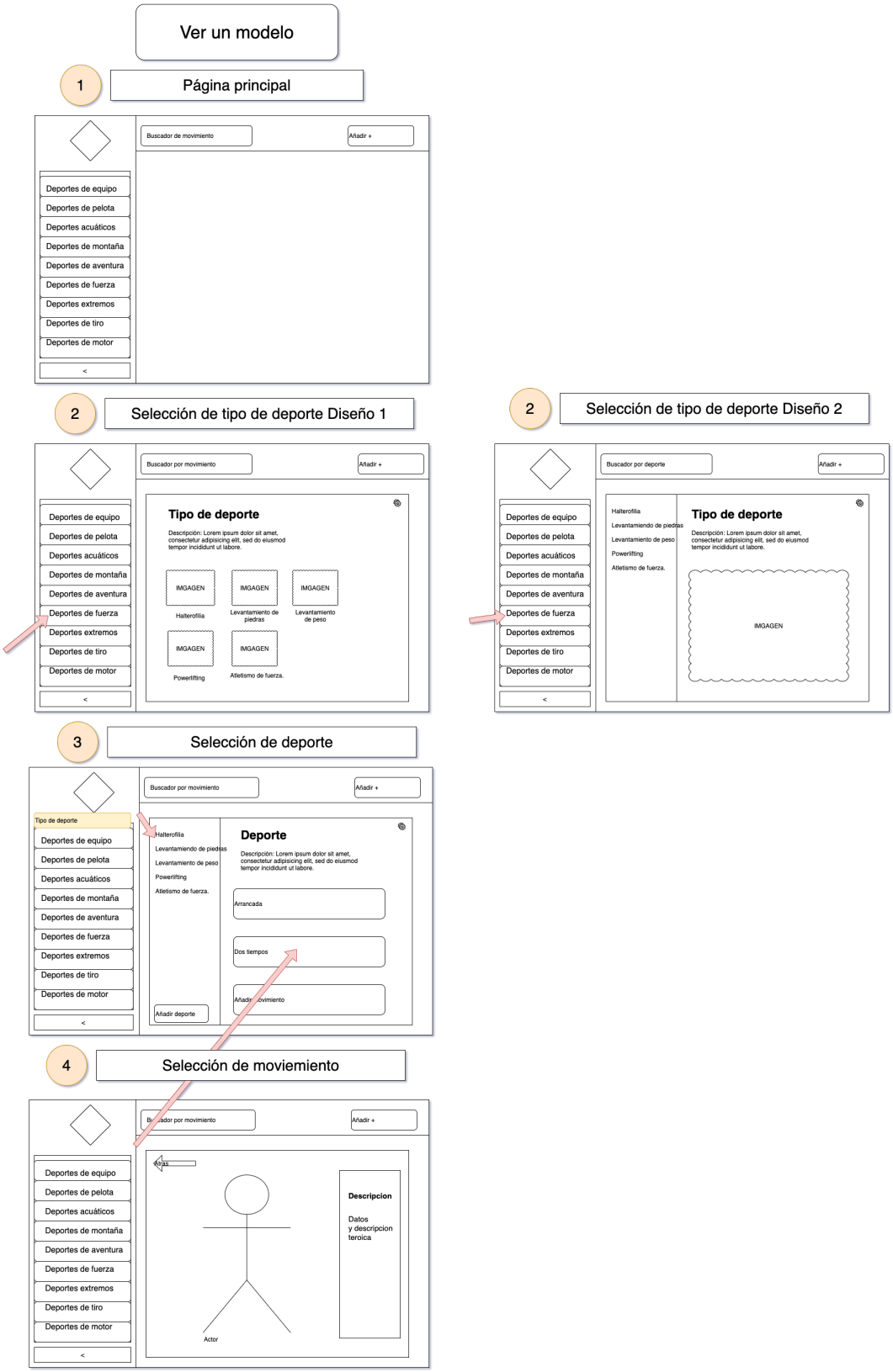
Quieres aportar alguna sugerencia o comentario?

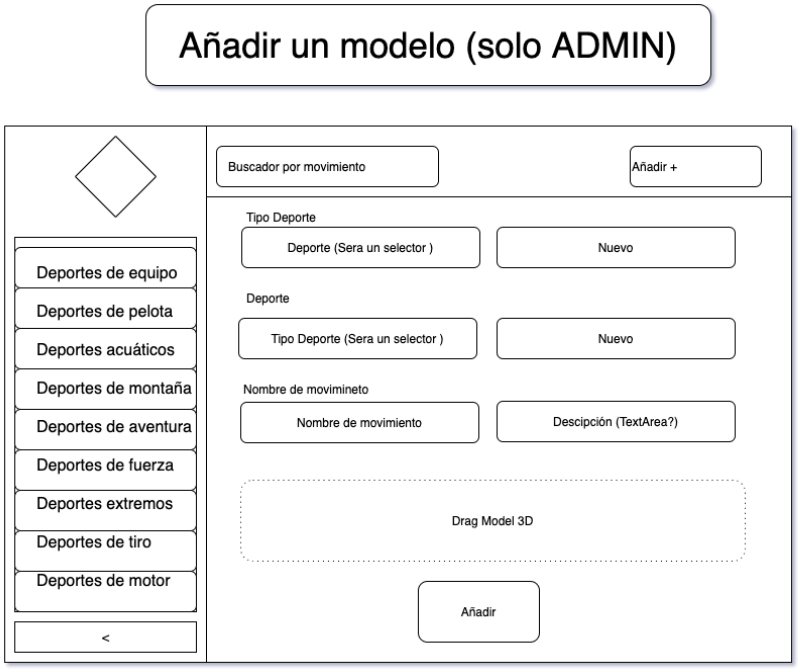
Texto de respuesta larga

Anexo V – Tareas Trello



Anexo VI – Diseño del front-end




Añadir un modelo (solo ADMIN)
Buscador por movimiento
Añadir +
Deportes de equipo
Deportes de pelota
Deportes acuáticos
Deportes de montaña
Deportes de aventura
Deportes de fuerza
Deportes extremos
Deportes de tiro
Deportes de motor
<
Tipo Deporte
Deporte (Sera un selector)
Nuevo
Deporte
Tipo Deporte (Sera un selector)
Nuevo
Nombre de movimineto
Nombre de movimiento
Descipción (TextArea?)
Drag Model 3D
Añadir

ANEXO VII – Swagger

TFG - From 3D movement to Biomechanical 1.0

[Base URL: /]
/api/v1/docs/swagger.json

The TFG API description

Schemes
HTTP

Authorize

Filter by tag

default

- GET /

auth

- POST /api/v1/auth/register Register a new user
- POST /api/v1/auth/login Log-in user
- POST /api/v1/auth/new-token Generates a new Access Token
- GET /api/v1/auth/user-data Get current logged user data
- POST /api/v1/auth/reset-password Reset Password
- POST /api/v1/auth/confirm-password-reset Confirm Reset Password
- GET /api/v1/auth/validate-user

roles

- GET /api/v1/roles Get all entities (paginated)
- POST /api/v1/roles Create a entity
- GET /api/v1/roles/{id} Get entity by ID
- DELETE /api/v1/roles/{id} Delete a entity by ID
- PUT /api/v1/roles/{id} Update a entity by ID

users

- GET /api/v1/users Get all entities (paginated)
- POST /api/v1/users Create a entity
- GET /api/v1/users/{id} Get entity by ID
- DELETE /api/v1/users/{id} Delete a entity by ID
- PUT /api/v1/users/{id} Update a entity by ID

sport

- GET /api/v1/sport Get all entities (paginated)
- POST /api/v1/sport Create a entity
- GET /api/v1/sport/{id} Get entity by ID
- DELETE /api/v1/sport/{id} Delete a entity by ID
- PUT /api/v1/sport/{id} Update a entity by ID

type-sport

- GET /api/v1/type-sport Get all entities (paginated)
- POST /api/v1/type-sport Create a entity
- GET /api/v1/type-sport/{id} Get entity by ID
- DELETE /api/v1/type-sport/{id} Delete a entity by ID
- PUT /api/v1/type-sport/{id} Update a entity by ID

movement

- GET /api/v1/movement Get all entities (paginated)
- POST /api/v1/movement Create a entity
- GET /api/v1/movement/{id} Get entity by ID
- DELETE /api/v1/movement/{id} Delete a entity by ID
- PUT /api/v1/movement/{id} Update a entity by ID

file-uploader

- POST /api/v1/file-uploader/multiple-files Upload multiple files
- POST /api/v1/file-uploader/avatar Upload image
- POST /api/v1/file-uploader/object Upload Object

Models

- AuthRegisterDto
- AuthLoginDto
- TokenDto
- NewTokenDto
- Role
- Profile
- User
- PasswordResetDto
- ConfirmPasswordResetDto
- Pagination
- RoleCreateDTO
- UserCreateDTO
- TypeSport
- Sport
- SportCreateDTO
- TypeSportCreateDTO
- Movement
- MovementCreateDTO
- FileResult